Írta:
**SZALKAI ISTVÁN és DÓSA GYÖRGY**

# ALGORITMIKUS SZÁMELMÉLET

**Egyetemi tananyag**


**2011**

LEKTORÁLTA: Dr. Hujter Mihály, Budapesti Műszaki és Gazdaságtudományi Egyetem Matematika Intézet Differenciálegyenletek Tanszék



TÁMOGATÁS:
Készült a TÁMOP-4.1.2-08/1/A-2009-0008 számú, „Tananyagfejlesztés mérnök informatikus, programtervező informatikus és gazdaságinformatikus képzésekhez” című projekt keretében.



ISBN 978 963 279 523 2

KÉSZÜLT: a Typotex Kiadó gondozásában

FELELŐS VEZETŐ: Votisky Zsuzsa

AZ ELEKTRONIKUS KIADÁST ELŐKÉSZÍTETTE: Bori Tamás

KULCSSZAVAK:

Algoritmusok, számelmélet, Euklidesz algoritmusa, Lineáris Diophantoszi egyenletek, kongruenciák, Kínai maradéktétel, prímtesztek, titkosírás nyilvános kulccsal, bizonyítás nulla információval, Euklideszi gyűrűk

ÖSSZEFOGLALÁS:

Az algoritmusok legfontosabb jellemzőinek áttekintése után a könyv bevezetésképpen az elemi számelméletet tekinti át algoritmikus szemszögből – nem minden egyszerű kérdés számítható ki percek vagy évek alatt. A könyv elsődleges célja egyrészt a számelmélet felhasználása az algoritmusok és titkosírások, irat-hitelesítések terén (7.3, 10. és 1. fejezetek), másrészt a számelmélet saját kiszámíthatósági problémái és azok megoldására történt próbálkozások bemutatása (3.2, 8. és 9. fejezetek). A könyv tartalmazza a fenti részek megértéséhez szükséges (nem kevés) elméleti tudnivalót is (2., 4–7. fejezetek). Az anyag megértését öt kis program segíti, melyek a bemutatott algoritmusokat lépésenként mutatják be, az output-szöveg szerkeszthető. A programok egyszerűbb számítások elvégzésére (pl. titkosírás) is használhatók. Néhány új jelöléssel és szemléletmóddal, továbbá Bolyai János számelméleti munkásságával, sőt modern algebrai általánosításokkal is megismerkedhetünk a könyvben (3., 8.3. és 13. fejezetek). A könyvet Táblázatok, Irodalomjegyzék és Tárgymutató teszi teljessé.

# Tartalomjegyzék

# 1. fejezet

# Bevezetés

„*Minden egész szám (lényegében) egyértelműen bontható fel prímszámok szorzatára*” – tanultuk általános iskolában, és fel is bontottunk néhány 3-4-jegyű számot.

**1.1. Példa.** Faktorizáljuk (bontsuk szorzótényezőkre) az alábbi számokat, vagy győződjünk meg róla, hogy prímszámok (vagyis nincs valódi felbontásuk):
**a)** $n_a = 440\,747$
**b)** $n_b = 2\,347\,589$
**c)** $n_c = 97\,189\,241$
**d)** $n_d = 17\,967\,876\,255\,379$
**e)** $n_e = 444\,113\,096\,135\,661\,846\,937$
**f)** $n_f = 2^{67} - 1 = 147\,573\,952\,589\,676\,412\,927$
**g)** $n_g = 11438162\,5757888867\,6692357799\,7614661201\,0218296721\,2423625625\,6184293570\,6935245733\,8978305971\,2356395870\,50589890751\,4759929002\,6879543541$ (129 jegyű).

Kedves Olvasónk, próbálja meg a fenti számokat faktorizálni (felbontani): kézzel (mint a XIX.században), egyszerű számológéppel vagy saját kis számítógépes programocskájával vagy könyvünkhöz mellékelt PRIM1D.EXE programmal (egyelőre ne használjon internetet, mint a 11. „*Számítógépes megvalósítások*” fejezetben), vagy olvassa végig könyvünket. (Most még a megfejtést se nézze meg a 3.25. Megoldásban a 2.2. „*A számelmélet algoritmikus problémái*” c. alfejezet végén.)

Igen, a bajok már a 8–10 jegyű számokkal elkezdődnek, pedig a modern alkalmazásokban többszáz vagy akár ezer jegyű egész- és prímszámokkal kellene számolnunk. Könyvünk lényegét a 2.2. „*A számelmélet algoritmikus problémái*” alfejezetben fejtjük ki részletesen: a *tényleges* számítások mennyi időt is igényelnek, hogyan csökkenthetők több évmillió (!) helyett pár napra. Ez vonatkozik egyrészt a számelméletben felmerülő számítási problémák (pl. prímtesztelés, -felbontás, lnko, lkkt, stb.) kiszámításának nehézségeire és azok megoldási módszereire, másrészt a számelmélet felhasználásaira a modern számítástechnikában (számítások gyorsításában, titkosírásokban, kódelméletben). Gyors algoritmus azonban nem létezik beható *elméleti* vizsgálatok nélkül, ezekből is csak a legszükségesebbeket tárgyaljuk (maradékosztályok, stb.).

A **titkosírások** *elvégezhetősége* azon alapszik, hogy aránylag könnyen találunk nagyméretű (500–1000 jegyű) prímszámokat (ld. 8. „*Prímkeresés*” fejezet) és aránylag könnyedén

tudunk számolni velük (ld. 5. „*Kongruenciák és maradékosztályok*” fejezet), míg *titkosságát* az biztosítja, hogy (jelenlegi ismereteink szerint) ugyanekkora, de ismeretlen számokat csak évezredekig tartó algoritmusokkal tudnánk prímtényezőkre bontani: ld. például a 7. „*Prímtesztelés és számok felbontása*” fejezet. (Egy egész számot akkor nevezünk „**ismeretlen**”-nek, ha nem ismerjük prímtényezős felbontását.)

Könyvünk mégis *bevezető jellegű*, hiszen csak néhány egyszerűbb szemléltető algoritmust mutat, és inkább csak hivatkozunk részletesebb művekre. (A téma legátfogóbb ismertetése még mindig **Donald Knuth** [KD] művében található.)

*Öt* egyszerű *számítógép-programot* is mellékeltünk könyvünkhöz:
EUKLDIO2D.EXE, HATVMODDD.EXE, KINAI3D.EXE, POLIOSZ5.EXE és PRIM1D.EXE. Nem díszes megjelenítés volt a célunk, hanem a könyvben leírt algoritmusok szemléltetése, lépésenkénti bemutatása. (Egyszerűségük miatt az adatok beírása sem „szerkesztősorban” történik: legyünk körültekintőek.) Jól használhatók azonban „számológép”-ként kisebb feladatok (pl. RSA) megoldásához és tanulmányozásához.

**A programok kizárólag magáncélra használhatók, bárminemű üzleti alkalmazásuk szigorúan tilos!**

Könyvünk feltételezi a középiskolás számelméleti anyag ismeretét, ugyanakkor az alapfogalmak *újszerű* bemutatásával (pl. $\mathfrak{p}(\mathbf{n})$, $\Delta$ és $\nabla$ jelek [3.9. és 3.33. Definíciók], atomelmélet és Boole-algebrák) igyekszik az anyag mélyebb megértését elősegíteni. Nem maradhattak ki a klasszikus és modern számelmélet legfontosabb és legérdekesebb problémái és eredményei sem, dióhéjban. A *Függelék*ben pedig az oszthatóság fogalmát és problémáit terjesztjük ki más halmazokra (Euklideszi gyűrűk), ezen vizsgálatok többek között a „Fermat-sejtés) megoldásában játszottak kulcsszerepet. (Az erre vonatkozó megjegyzéseinket a Függeléken kívül apróbetűvel jeleztük.)

Köszönetünket fejezzük ki kedves tanárainknak: **Szalay Mihály**nak, **Freud Róbert**nak, és **Csirmaz László**nak! Külön köszönet a **Lektor** lelkiismeretes munkájának!

## 1.1. Jelölések

A könyvben használt legfontosabb **jelölések** a következők:

$\mathbb{N}$ jelöli a **természetes** számok halmazát, azaz

$$\mathbb{N} := \{0, 1, 2, \dots\},$$

az **egész** számok halmazát $\mathbb{Z}$-vel jelöljük.

$\mathbb{P}$ jelöli a prímszámok halmazát.

$\#\{\dots\}$ vagy $|\{\dots\}|$ a halmaz számossága,

$\mathfrak{p}(n) := n$ prímosztóinak *multihalmaza* ($n \in \mathbb{N}$), pl. $\mathfrak{p}(12) = \{2, 2, 3\}$

$\mathrm{int}(x) = \lfloor x \rfloor$ = *(alsó)* **egészrész-függvény**: a $x$-nél nem nagyobb egész számok közül a legnagyobb („lefelé csonkítás” a nemnegatív számok esetén),

$\lceil x \rceil$ = **felső egészrész-függvény**: a $x$-nél nem kisebb egész számok közül a legkisebb („felfelé kerekítés” a nemnegatív számok esetén).

*Másképpen*: minden $x \in \mathbb{R}$ valós számra $\lfloor x \rfloor, \lceil x \rceil \in \mathbb{Z}$, $\lfloor x \rfloor \leq x \leq \lceil x \rceil$ és egyenlőség csak $x \in \mathbb{Z}$ egész számoknál van.

Tudjuk, hogy egy (véges vagy végtelen) *sorozat* semmi esetre sem részhalmaz, de kényelmesek az $(a_n) \subset \mathbb{N}$, $(a_n) \subset \mathbb{R}$ ill. $(m_1, m_2, \ldots, m_k) \subset \mathbb{R}$ jelölések.

Tudjuk azt is, hogy sok számítógép-program tizedes pontot használ, mi mégis maradunk a *tizedes/ bináris-* **vessző**nél, hiszen Magyarországon a tudományos- és közéletben, oktatásban és tankönyvekben (és a Windows-rendszerben is) ez az elterjedt.

Néhány absztrakt matematikai fogalmat (Boole Algebrák, gyűrűk, testek stb.) a *Függelékben* vázlatosan ismertetünk.

□ egy-egy gondolat / Definíció / Megjegyzés / Állítás / Tétel / Bizonyítás végét jelöli.

# 2. fejezet

# Algoritmusok sebessége

## 2.1. Alapfogalmak

Az algoritmus fogalmát precízen lehet definiálni (pl. [SzI2]), nekünk elég a következő intuitív magyarázat: „*a probléma megadása után a számítógép* **véges idő** *után* **megáll***, és* **helyes** (pontos) **eredmény***t ad*”, és természetesen „*ugyanazon inputtal megismételt minden futás ugyanazt az eredményt adja*” – az ilyen algoritmusokat **determinisztikus algoritmus**oknak hívják. Vizsgálhatnánk még **nemdeterminisztikus algoritmus**okat is: ugyanarra az inputra nem minden esetben áll meg és nem mindig ad helyes eredményt – ilyen algoritmusokkal könyvünkben nem foglalkozunk. Vizsgálni fogunk azonban olyan algoritmusokat, melyek minden futás után megállnak (véges idő után), de néha csak olyan típusú választ kapunk tőlük: „*a vizsgált problémára 99% eséllyel az a válasz hogy . . .* ”, esetleg még az ilyen típusú válaszok valószínűségét is meg tudjuk becsülni. Az ilyen algoritmusokat **valószínűségi algoritmusok**nak nevezzük. A valós számoknál megismert **közelítő algoritmusok**at is csak elvétve használjuk könyvünkben.

A determinisztikus algoritmusoknál említett ”*véges idő*” elég tág fogalom: $10^{100}$ lépés (1000 GHz = $10^{12}$ lépés/másodperc sebességgel) is véges, ez pedig „csak” $3 \times 10^{80}$ év . . . , márpedig a titkosírásoknál, többszázjegyű számoknál látni fogunk olyan problémákat, amelyekre ennél gyorsabb algoritmust (2010-ben) még senki sem ismer.

Az algoritmusok futásának gyorsaságát, időigényét az algoritmus (pontosabban az általa megoldott *probléma*) *bonyolultságának* nevezünk.

Az algoritmusok futási idejének kiszámításakor teljesen pontos számításra nincs szükségünk, hiszen ha mondjuk hónapokig futott, további pár óra már nem érdekes. Ezenkívül, programunk sok egyéb műveletet végez (pl. beolvasás, kiírás, stb), másrészt, az egyre gyorsabb számítógépek megjelenése miatt (ugyanaz a program sokkal gyorsabban futhat egy másik számítógépen), és részben ugyanezen okok miatt bennünket inkább a nagyméretű adathalmazok érdekelnek. Vagyis csak az idő nagyságrendjére, aszimptotikus viselkedésére vagyunk kíváncsiak.

Sajnos az alábbi jelölések nem egységesek, az idők folyamán is változtak, mi az alábbiakat használjuk ([CLR]):

**2.1. Definíció.** *Legyenek* $f, g\colon \mathbb{N} \to \mathbb{R}^+$ *tetszőleges pozitív értékű függvények. Azt mondjuk, hogy*

**(i)**

$$f = \mathcal{O}(g) \tag{2.1}$$

*vagyis* **f értéke nagy ordó g** *("***f is big oh of g***"), ha valamely fix* $c_2 \in \mathbb{R}^+$ *számra teljesül a*

$$f(n) < c_2 \cdot g(n) \tag{2.2}$$

*egyenlőtlenség minden elég nagy* $n \in \mathbb{N}$ *számra, azaz létezik olyan* $n_0 \in \mathbb{N}$ küszöbszám: (2.2) teljesül minden $n > n_0$ természetes szám esetén,

**(ii)**

$$f = \Theta(g) \tag{2.3}$$

*ha valamely fix* $c_1, c_2 \in \mathbb{R}^+$ *számokra teljesül a*

$$c_1 \cdot g(n) < f(n) < c_2 \cdot g(n) \tag{2.4}$$

*egyenlőtlenség minden elég nagy* $n \in \mathbb{N}$ *számra,*

**(iii)**

$$f \in \Omega\left(g\right) \tag{2.5}$$

*ha valamely fix* $c_1 \in \mathbb{R}^+$ *számra teljesül a*

$$c_1 \cdot g(n) < f(n) \tag{2.6}$$

*egyenlőtlenség minden elég nagy* $n \in \mathbb{N}$ *számra,*

**(iv)**

$$f = o\left(g\right) \tag{2.7}$$

*vagyis* **f értéke kis ordó g** *("***f is little oh of g***"), ha* minden $c_2 \in \mathbb{R}^+$ *számra teljesül a*

$$f(n) < c_2 \cdot g(n) \tag{2.8}$$

*egyenlőtlenség minden elég nagy* $n \in \mathbb{N}$ *számra, vagy másképpen:*

$$\lim_{n\to\infty} \frac{f\left(n\right)}{g\left(n\right)} = 0,$$

**(v)**

$$f = \omega\left(g\right) \tag{2.9}$$

*ha* minden $c_1 \in \mathbb{R}^+$ *számra teljesül a*

$$c_1 \cdot g\left(n\right) \leq f\left(n\right) \tag{2.10}$$

*egyenlőtlenség minden elég nagy* $n \in \mathbb{N}$ *számra, vagy másképpen:*

$$\lim_{n\to\infty} \frac{f\left(n\right)}{g\left(n\right)} = \infty. \tag{2.11}$$

□

(Lényegében az $\mathcal{O}$ és $\Theta$ jelölések a „*körülbelül*” szó matematikai szinonimái.)

Természetesen a fenti jelöléseknek csak akkor van értelme, ha $f$ bonyolult és $g$ egyszerű. A képletben szereplő $c_1, c_2$, (tetszőleges) konstansok „nyelik le” a bevezetőben említett „kerekítési” hibákat ill. a különböző sebességű gépek problémáját ahol $n$ az input mérete, és $n \to \infty$. A megkövetelt $n_0$ küszöbszámok értékei lényegtelenek, hiszen mi az algoritmusokat „tetszőlegesen nagy” inputokra tervezzük, matematikai értelemben $n \to \infty$.

Az analízisből is jólismert típusok ($g$ lehetséges értékei) esetén a következő elnevezések használatosak (növekvő sorrendben):

**2.2. Definíció.** $f = \Theta(g)$ vagy $f = \mathcal{O}(g)$ esetén az $f$ és $g$ függvényeket az alábbi elnevezésekkel illetjük:

$g(n) = c$ – **konstans** (inputtól független idő),

$g(n) = \log_a(n)$ – **logaritmikus** (alap tetszőleges de 1-nél nagyobb),

$g(n) = n$ – **lineáris (elsőfokú)**,

$g(n) = n \cdot \log_a(n)$ – **szemilineáris** (alap tetszőleges),

$g(n) = n^2$ – **négyzetes** (**kvadratikus**),

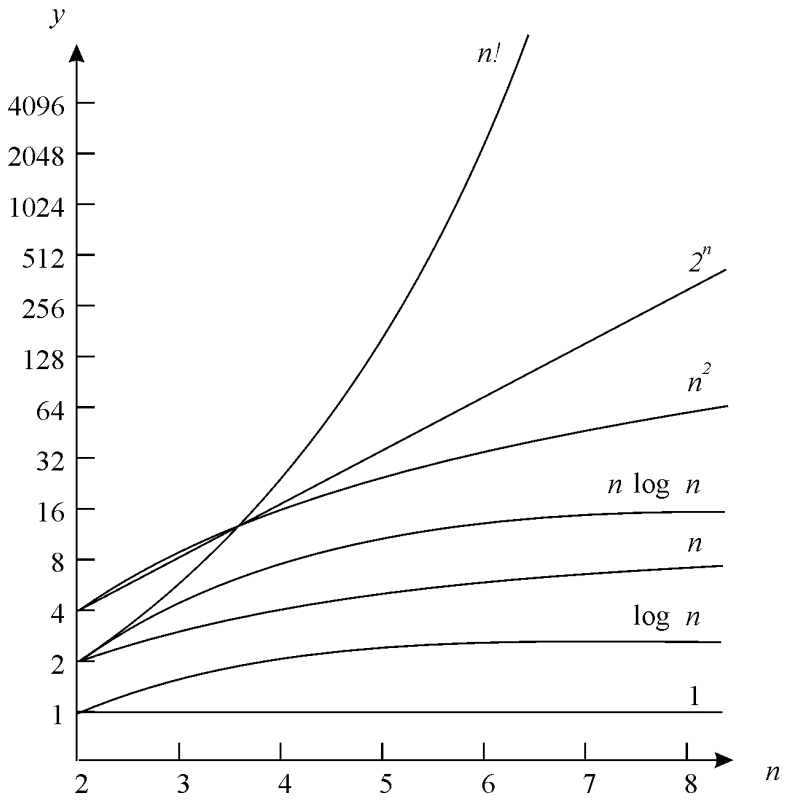
$g(n) = n^k$ – **polinomiális** ($k > 1$, $k \in \mathbb{R}$ tetszőleges de rögzített valós szám),

$g(n) = a^n$ – **exponenciális** (alap mindegy de 1-nél nagyobb),

$g(n) = n^n$ – **hiperexponenciális.** □

**2.3. Megjegyzés.** Mint középiskolában tanultuk: *bármely* $a, b \in \mathbb{R}^+$ esetén $\log_b(x) = c \cdot \log_a(x)$ egy rögzített $c \in \mathbb{R}^+$ számra ($c = \frac{1}{\log_a(b)}$), vagyis $\mathcal{O}$ és $\Theta$ tekintetében a logaritmus alapja lényegtelen. □

Érdemes az alapfüggvényeket közelebbről is megvizsgálnunk.




*Az alapfüggvények*

Az ábrán a jólismert alapfüggvényeket ábrázoltuk, de gyors növekedésük miatt függőleges összenyomást alkalmaztunk, ráadásul logaritmikus beosztásút, azaz a függőleges összenyomás mértéke távolabbi síktartományok esetén egyre növekszik. Ennek következtében a lineáris (egyenes) és a parabola görbék most lefelé görbülnek, az exponenciális görbe kiegyenesedett (vagyis az exponenciális függvény éppen akkora mértékben növekszik, mint amekkora az összenyomás mértéke). Azonban a hiperexponenciális görbék még most is meredeken ívelnek felfelé: még ekkora összenyomás ellenére is csak $n < 7$ (hét) értékek esetén férnek el a papíron!

Nagyobb $n$ értékekre csak számításokkal tudjuk már az exponenciális függvényeket is nyomon követni, [SzI2] Függelékében, vagy a szerzők honlapján levő táblázatban:

HTTP://MATH.UNI-PANNON.HU/~SZALKAI/ALGTABL.PDF

egy 1 MHz-cel éjjel-nappal működő számítógép konkrét futásidejét számítottuk ki $n = 10, 100, 10, \ldots, 10^6$ méretű adathalmazok esetén. Érdemes a „több évmilliócska” időnél jóval nagyobb időket tanulmányozni, amikor is $100 - 1000 - 10^6$-szor (vagy esetleg négyzetesen?!) gyorsabb számítógépek illetve programok beszerzése ugye nem sokat segít ....

Analízisben jól ismertek az alábbi összefüggések, amelyek igazolják a 2.2. Definíció elnevezéseinek sorrendjét:

**2.4. Tétel.**

$$\log_a(n) \ll n^k \ll b^n \ll n^n \qquad \left(a, b, k \in \mathbb{R}^+ ,\ 1 < b\right)$$

*azaz*

$$\lim_{n\to+\infty} \frac{\log_a n}{n^k} = 0, \quad \lim_{n\to+\infty} \frac{n^k}{b^n} = 0, \quad \lim_{n\to+\infty} \frac{b^n}{n^n} = 0 \qquad \left(a, b, k \in \mathbb{R}^+,\ 1 < b\right)$$

□

A fenti tételben és a továbbiakban az alábbi általános jelöléseket használjuk:

**2.5. Definíció.** Tetszőleges $f, g\colon \mathbb{N} \to \mathbb{R}$ vagy $f, g : \mathbb{R}^+ \to \mathbb{R}$ függvényekre
**(i)** az

$$f \ll g$$

jelölést akkor használjuk, ha $g$ **végtelenszer nagyobb**, mint $f$, vagyis

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \quad \text{illetve} \quad \lim_{x\to\infty} \frac{f(x)}{g(x)} = \infty.$$

**(ii)**

$$f \sim g$$

azt jelöli, hogy $f$ és $g$ **aszimptotikusan egyenlőek**, vagyis

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 \quad \text{illetve} \quad \lim_{x\to\infty} \frac{f(x)}{g(x)} = 1. \quad \square$$

Az (ii)-beli jelölést ne tévesszük össze a gyűrűknél használatos "*asszociált elemek*" jelöléssel (ld. Függelékben).

Ne feledjük a kombinatorikában (és jegyzetünkben is) gyakran használt közelítő formulát:

**2.6. Tétel** (Stirling-formula)**.** *Elég nagy $n \in \mathbb{N}$ természetes szám esetén*

$$n! \sim \left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n} \tag{2.12}$$

*sőt kicsit pontosabban*

$$\left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n} \cdot e^{\frac{1}{12n} - \frac{1}{360n^3}} \le n! \le \left(\frac{n}{e}\right)^n \cdot \sqrt{2\pi n} \cdot e^{\frac{1}{12n}}. \tag{2.13}$$

*Más alakja:*

$$\log(n!) \sim n \cdot \log(n) - n. \quad \square$$

*(James Stirling (1692-1770) skót matematikus.)*

Hangsúlyozzuk ismét, hogy számunkra $n \to \infty$ miatt a konstansok nem érdekesek. Ezért (elméletileg) például a $54\ 312 \cdot n^2 = \mathcal{O}(n^2)$ (négyzetes) idejű algoritmust jobbnak tartjuk mint a $10^{-3} \cdot 2^n = \mathcal{O}(2^n)$ (exponenciális) idejűt, noha kis $n$ értékekre (elég sokáig) az utóbbi

a gyorsabb, vagyis gyakorlati felhasználásra kis $n$-ek esetén inkább az utóbbi javasolt (de nagy $n$-re már nem).

A „tanulság”: az $n^2$ algoritmusokat még jónak, a polinomiális algoritmusokat még elfogadhatónak („gyorsnak”), az exponenciálisakat pedig (kibírhatatlanul) lassúnak, a hiperexponenciálisakat pedig . . . -nak tartjuk.

Egy konkrét algoritmus futásidejét aránylag egyszerű kiszámolni, azonban fontosabb *lenne* az algoritmusok (pontosabban: a problémák) bonyolultságának *alsó* becslése: mennyire *szükségszerűen* nehéz egy-egy probléma. Sajnos ez nagyon nehéz, általában megoldatlan feladat, kevés (és nehezen bizonyítható) ilyen eredményünk van.

Most csak pár érdekességet említünk meg:

**2.7. Tétel** (Cejtin)**.** *Tetszőleges $f : \mathbb{N} \to \mathbb{N}$ rekurzív (képlettel kiszámítható) függvényhez létezik olyan probléma, melyet megoldó minden algoritmus, $n$ adat esetén legalább*

$$\mathcal{O}(f(n))$$

*ideig fut.*

**Bizonyítás.** Ötlet: ha már a végeredmény kijelzéséhez legalább ennyi betűt kell leírnunk . . . . ■

(A rekurzív függvények definíciója megtalálható pl. [SzI2]-ben.)

**2.8. Definíció.** (Naiv=nem matematikai): *Egy problémát* **NP-teljes***nek* (**Nondeterministic Polynomial Complete**) *nevezünk, ha az alábbi kijelentés igaz rá:*

*” ha erre a problémára lenne gyors algoritmus, akkor a világ összes problémájára is lenne (azonnal!) gyors algoritmus ” .* □

Bármilyen hihetetlen: már a múlt század 70-es éveiben közismert „hétköznapi” problémák tucatjairól bizonyították be, hogy NP -teljes (ld. pl. [SzI2], [LG], [CLR] vagy [LL2]).

Algoritmusok elméletéről rövid bevezetőt találunk [SzI2] utolsó 40 oldalán, nagyon alapos de olvasmányos bevezető mű [LG], részletesebb olvasnivalót például [CLR], [IA], [LL2], [RF] és [RISz]-ban találhatunk.

## 2.2. Alapműveletek sebessége

### 2.2.1. Természetes számok mérete

Az algoritmus futásideje nyilván függ az Input méretétől: mekkora adathalmazt adtunk a számítógépnek. Ne feledjük:

**2.9. Megjegyzés.** Amennyiben az input egy $n \in \mathbb{N}$ természetes szám, akkor az input mérete = az $n$ *számjegyeinek száma*, tehát nem $n$ értéke maga.
Az input méretét néha $\mathbf{in}$-nel vagy $\mathbf{in}\,(n)$-nel jelöljük, azaz $n > 0$ esetén

$$\mathbf{in}(n) := \lfloor \log(n) \rfloor + 1$$

(a számrendszer alapszáma, és így a logaritmus alapja is ugye lényegtelen). □

**2.10. Példa.** Ha a számítógépnek egy $k$ (pl. $k = 100$) jegyű $n$ számot adunk be, akkor mi csak $k$ db karaktert adtunk a gépnek, de ez a karaktersorozat által kódolt $n$ szám értéke valójában $10^k$ körül van: $n = \mathcal{O}\left(10^k\right)$. Ha most ezt az $n$ számot szorzattá akarjuk bontani, például megkérhetjük a gépet, hogy a páratlan számokkal próbálja meg $n$-et maradék nélkül elosztani $\sqrt{n}$-ig, akkor a végrehajtott $\mathcal{O}\left(\sqrt{n}\right)$ lépés valójában $\mathcal{O}\left(10^{k/2}\right) = \mathcal{O}\left(\sqrt{10}^k\right)$, vagyis inputunk függvényében *exponenciálisan* lassú algoritmus!
(Az már elenyésző probléma, hogy egy-egy lépésben többszázjegyű egész számokat kell maradékosan elosztanunk egymással.) □

**2.11. Definíció.** Legyen $n \in \mathbb{N}$ tetszőleges természetes szám. Ekkor jelölje

$$\ll n \gg := \lfloor \log_2(n) \rfloor + 1 \tag{2.14}$$

az $n$ szám bináris számjegyeinek számát. Néha használjuk a $k_b$ és $k_d$ jelöléseket is az $n$ szám bináris illetve decimális számjegyinek számára, ebben az esetben az $n$ számot ***k*-bites szám**nak nevezzük. □

Mint az 2.3. Megjegyzésben is említettük, a (2.14)-ben a logaritmus alapszáma lényegtelen, ezért a „*k-bites szám*" elnevezésben is lényegtelen (elméletileg) a számrendszer alapszáma, mindenképpen $\ll n \gg = \mathcal{O}(\log(n))$, és egy $k$-bites szám *értéke* már $10^{k_d}$ ill. $2^{k_b}$: exponenciálisan nagy.

[LL1]-ban az $n$ szám (kettes számrendszerbeli) jegyeinek számát $\langle n \rangle$ jelöli, mi helyette $\ll n \gg$-t használunk amiatt, hogy ne legyen összetéveszthető más jelölésekkel, például az *Euklideszi algoritmus*nál az $\langle r_i \rangle$ maradékok megkülönböztetésére.

[JA] 34. oldala szerint a számrendszer alapja a *gyakorlatban* mégsem lényegtelen: „*a Maple* $10^4$ *alapú számrendszert használ, ami lényegesen lerontja a sebességet*".

**2.12. Példa.** Az alábbi számokról döntsük el, hogy hány számjegyből állnak a 10-es számrendszerben:

$a = 2^7 \cdot 3^{43} \cdot 11^{93} \cdot 39^{45} \cdot 101^2$,

$b_{1,2} = 697\,053\,813 \cdot 2^{16\,352} \pm 1$ (ikerprímek,[JA]),

$c_{1,.2} = 242\,206\,083 \cdot 2^{38\,880} \pm 1$ (ikerprímek,[JA]),

$d_{1,.2} = 16\,869\,987\,339\,975 \cdot 2^{171\,960} \pm 1$ (ikerprímek,[FR]),

$M_{43\,112\,609} := 2^{43\,112\,609} - 1$ (egy Mersenne-prím),

$M_{\ddot{o}} = 2^p - 1$ ahol $p = 2375063906985 \cdot 2^{19380}$
(egy **M**ersenne **ö**sszetett szám[JA]).

Hány oldalon illetve hány (kilo)méter polcon férne ki $M_{\ddot{o}}$ ha 4 pt betűméretben (152 sor, soronként 225 karakter, „biblia"-papíron 1500 lap = 4 cm) nyomtatnánk ki?

**2.13. Megoldás.** Egy $n \in \mathbb{N}$, $0 < n$ szám $k$-alapú számrendszerben felírt jegyeinek számát nyilván

$$\lfloor \log_k (n) \rfloor + 1 \tag{2.15}$$

adja meg.

Tehát

$\lg(a) = 7\lg(2) + 43\lg(3) + 93\lg(11) + 45\lg(39) + 2\lg(101) \approx 195,0794$
miatt az $a$ szám 196 jegyű,

$\lg(b_1) \approx \lg(697053813) + 16352\lg(2) \approx 4931,285\,755$
miatt a $b_1$ és $b_2$ számok 4932 jegyűek,

$\lg(c_{1,2}) \approx \lg(242\,206\,083) + 38880\lg(2) \approx 11\,712,430\,416$
miatt a $c_1$ és $c_2$ számok 11 713 jegyűek,

$\lg(d_{1,2}) \approx \lg(16\,869\,987\,339\,975) + 171\,960 \cdot \lg(2) \approx 51778,345$
miatt a $d_1$ és $d_2$ számok 51 779 jegyűek, ([FR] szerint ezek voltak 2005-ben a legnagyobb ismert ikerprímek, Járai Antal és munkatársai találták meg),

$\lg(M_{43\,112\,609}) \approx 43\,112\,609\lg(2) \approx 12\,978\,188,5003329$
miatt az $M_{43\,112\,609}$ szám 12 978 189 jegyű
(tizenkétmillió számjegyű prímszám!),

$\lg(M_{\ddot{o}}) \approx 2^{2375063906985 \cdot 2^{19380}} \cdot \lg(2)$-t nagy mérete miatt már a Windows számológépe sem tudja kiszámítani, ezért $\lg(\lg(M_{\ddot{o}}))$ -t számoljuk ki:

$$\lg(\lg(M_{\ddot{o}})) \approx \lg\left(2^{2375063906985 \cdot 2^{19380}} \cdot \lg(2)\right) =$$
$$= 2375063906985 \cdot 2^{19380} \cdot \lg(2) + \lg\lg(2) \approx 6,540 \times 10^{5845}$$

vagyis $M_{\ddot{o}}$ számjegyeinek *száma* $\approx 10^{5845}$. Csak ahhoz kell 5845 számjegy, hogy leírjuk: „$M_{\ddot{o}}$-nek ennyi számjegye van"!
$M_{\ddot{o}}$ számjegyeinek leírásához $10^{5845}$ karaktert kell leírnunk, a papírlapokat szorosan egymáshoz téve kb. $8,2 \times 10^{5816}$ *fényévnyi* polcra lesz szükségünk (ez nem 8 fényév, hanem több, mint $10^{5816}$ fényév!)

Szerencsére, ebben a könyvben mi nem foglalkozunk ekkora számokkal.

### 2.2.2. Műveletek sebessége

Mivel a modern titkosírásoknál többszázjegyű számokkal kell exponenciálisan sok (!) műveletet végeznünk, érdemes gondolkodnunk az alapműveletek gyorsabb elvégzésén.

#### ÖSSZEADÁS és KIVONÁS

Az (általános) iskolában tanult „papír-ceruza” módszer lineáris: ha az $m \pm n$ feladatban mondjuk $n$ a nagyobbik abszolút értékű, és

$$k := in := \ll n \gg = \lfloor \log_2 (n) \rfloor + 1$$

akkor az algoritmus *lépésszáma* $\leq 2 \cdot in$. $\square$

Lineárisnál gyorsabb algoritmus (számok összeadására) nyilván nem létezik.

#### SZORZÁS

A tanult írásbeli szorzás kvadratikus (négyzetes), azaz a *lépésszám* =

$$\mathcal{O} (\ll n \gg \cdot \ll m \gg) \leq 2 \cdot in^2$$

([KRSz] 105.old., [KN] 7.old.).

Azonban Lovász László és Gács Péter [LG] nagyszerű könyvének 3.1.1.fejezetéből (81-82.old.) megtanulhatjuk *Karacuba* szovjet matematikus 1962-ben felfedezett szorzási módszerét: két $k$ jegyű számot módszerével $k^2$ lépés helyett

$$27 \cdot k^{\log_2(3)} \approx 27 \cdot k^{1,85}$$

lépésben szorozhatunk össze! (Az algoritmus lényege: megpróbál 2 hatványaival szorozni, ami pedig lényegében a *binárisvesszőtologatásának* felel meg. Az algoritmust röviden megemlíti [JA] is.)

[LG] még megemlíti, hogy Karacuba módszerét azóta messzemenően általánosították *Schönhage és Strassen* svájci matematikusok: a véges Fourier-transzformáltak felhasználásával csak

$$c \cdot k \cdot \log (k) \cdot \log (\log (k))$$

műveletet igényel két $k$ jegyű szám összeszorzása. (Ld. még [KD] 2.kötet 4.3.3. alfejezetét.)

#### OSZTÁS

Az iskolában tanult algoritmus (egész számok maradékos osztása) polinomiális ugyan, de négyzetesnél is több időt igényel. [CLR] (692, 696.old., 33.1-11.gyakorlat) és [KRSz] 7.old. szerint négyzetes algoritmus is könnyen szerkeszthető. Kissé bonyolultabban, egy egyszerű „oszd meg és uralkodj” elven működő módszerrel egy $k$-bites számot egy nála kisebbel el lehet osztani $\mathcal{O}\left(k^{\lg_2(3)}\right)$ lépésben is, sőt a leggyorsabb ismert módszer $\mathcal{O} (k \log (k) \log (\log (k)))$ futásidejű. Gyakorlati célokra azonban a $\mathcal{O} (k^2)$ algoritmus a legjobb (egyszerűsége végett).

A valós számokra készített *közelítő* algoritmusokat is használhatjuk – a hibakorlátot állítsuk $\frac{1}{2}$-nél kisebbre. A már említett [LG] könyv 3.2.1. fejezetében (90-91old.) megismerthetjük *Newton* módszerét: ha $u \in \mathbb{R}$ és $1/u$-t $\ell$ értékes jegyig akarjuk kiszámolni, akkor $c \cdot \ell \log(\ell) \log(\log(\ell))$ műveletre van szükségünk. [JA] a Newton-féle iterációt javasolja: $1/u$ kiszámításához az $f(x) = u - \dfrac{1}{x}$ függvény zérushelyét keressük iterációval: legyen például $x_0 := 1$ és $n \in \mathbb{N}$ esetén

$$x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{u - \frac{1}{x_n}}{\frac{1}{x_n^2}} = 2 \cdot x_n - u \cdot (x_n)^2 . \tag{2.16}$$

### HATVÁNYOZÁS

Nyilvánvalóan pl. a $2^n$ végeredmény puszta kiírásához is már $\log(2^n) = n = 2^{in}$ lépés kell, de a részletszámítások leírása sem elhanyagolható feladat. Még akkor sem, ha ismételt négyzetreemelésekkel a lépések számát lényegesen csökkentjük.

Azonban, ha egy $u^k$ hatványnak csak egy $m$ számmal vett *maradékára* van szükségünk, akkor a fenti trükk már szemilineáris algoritmust eredményez, amint ezt a 6.6. "*Nagy kitevőjű hatványozás*" alfejezetben részletesen ismertetjük.

### FAKTORIÁLIS

A szokásos módszer $n!$ kiszámításához $\mathcal{O}\left(n^2 \cdot \log^2 n\right) = \mathcal{O}\left(2^{2 \cdot in} \cdot in^2\right)$ lépést igényel, ennél lényegesebben jobb algoritmus nem ismert.

### NÉGYZETGYÖKVONÁS

Nyilván $\sqrt{n}$ egészrészére vagyunk kíváncsiak ha $n \in \mathbb{N}$. A "*kb. fele olyan hosszú*" algoritmus is jó, de nem a leggyorsabb. Egyik lehetőség ismét egy Newton-féle iterációs módszer: legyen például $x_0 := 1$ és $n \in \mathbb{N}$ esetén

$$x_{n+1} := \frac{x_n + \frac{n}{x_n}}{2} .$$

### KONVERTÁLÁS

Mint már idéztük [JA]-t: néha célszerű a számrendszer alapszámát megváltoztatnunk. Használhatjuk az iskolai módszert is, kicsit jobb [KN] 7.old. módszere: $k$-bites bináris számot átír tízes számrendszerbe $\mathcal{O}(k^2)$ idő alatt. (A konvertálás fontosságáról és különböző megoldási módszereiről részletesebben olvashatunk [KD] 2.kötet 4.4. alfejezetében.)

# 3. fejezet

# A számelmélet alapjai

A középiskolai számelmélet-ismereteket (oszthatóság, prímszámok, prímfelbontás, *lnko*, *lkkt*) ismertnek tételezzük fel. Most csak felsoroljuk a legfontosabb fogalmakat és összefüggéseket, a hangsúlyt inkább az *új szemléletre* helyezzük (pl. a $\mathfrak{p}\,(\mathbf{n})$, $\Delta$ és $\nabla$ jelek a 3.9. és 3.33. Definíciókban, hasonlat a kémiai atomok elméletével, Boole-algebrák, stb.) Ezenkívül néha utalunk (apró betűkkel) az oszthatóság általánosabb (más halmazokban is definiálható) tulajdonságaira, amiket részletesebben a *Függelék*ben ismertetünk.

Ismét hangsúlyozzuk, hogy a könyvben „szám" alatt legtöbbször *egész számot* értünk.

## 3.1. Oszthatóság és prímszámok

Egész számok osztásakor természetes, hogy van maradék. Most csak a „nem maradt" esettel foglalkozunk, későbbi fejezetekben már inkább maga a maradék lesz fontosabb a hányadosnál.

**3.1. Definíció.** Legyenek $a, b \in \mathbb{Z}$ egész számok. Azt mondjuk, hogy $a$ **osztója** $b$**-nek**, vagy más szóval, $b$ **osztható** $a$ **-val**, ha létezik olyan $x \in \mathbb{Z}$, hogy $b = ax$. Ennek jelölése $a \mid b$. □

**3.2. Definíció.** A $p \in \mathbb{Z}$ egész számot ($p \neq -1,\ 1,\ 0$) **prímszám**nak (=primitív szám) röviden **prím**nek, vagy törzsszámnak nevezünk, ha **irreducibilis** (**felbonthatatlan**), azaz nem írható fel $p = x \cdot y$, $x, y > 1$ alakban.
Másképpen fogalmazva: $p$-nek nincs olyan $d$ osztója, amelyre $1 < d < p$, azaz önmagán és az 1-en kívül nincs más pozitív osztója.
Ha az $n$ egész szám nem prím, akkor **összetett szám**nak nevezzük.
A *pozitív* prímszámok halmazát $\mathbb{P}$-vel jelöljük. □

**3.3. Megjegyzés.** **(i)** A fenti definícióból ki kellett zárnunk $\pm 1$-et és $0$-át, mert egészen más tulajdonságokkal rendelkeznek mint a prímszámok. A prímszámok legfontossabb tulajdonsága *nem* felbonthatatlanságuk, hanem az, hogy minden egész szám előállítható belőlük (ld. 3.6. Tétel).
**(ii)** Egész számok oszthatóságánál az előjel nyilván nem lényeges, de mindig gondolnunk kell a negatív egész számokra is, hiszen azok is léteznek. Így például minden $p \in \mathbb{P}$ esetén $-p$ is prímszám!

**3.4. Megjegyzés.** Az oszthatóság fogalmát – és innen az egész könyv elméletét is! – általánosíthatjuk tetszőleges gyűrűkre is, pl. polinomokra, komplex (algebrai) egészekre, Gauss- és Euler-egészekre, mátrixokra. Rövid bevezetőt találunk a Függelékben vagy [KN]-ben. □

**3.5. Tétel.** *Ha $p$ prím és $p \mid ab$, akkor vagy $p \mid a$ vagy $p \mid b$. (Ezt **prímtulajdonság**nak hívjuk). Ugyanígy, ha $p$ prím és $p \mid a_1 a_2 \cdots a_n$, akkor valamely $i \leq n$-re $p \mid a_i$.* □

Hangsúlyozzuk, hogy a prímszámok jelentőségét nem az előző definíció, hanem a *következő tétel* magyarázza meg:

**3.6. Tétel** (Számelmélet Alaptétele)**.** *Minden $n \in \mathbb{Z}$, $n \neq 0$ egész szám felbontható ("**faktorizálható**") prímszámok szorzatára, lényegében egyértelműen (azaz csak a tényezők sorrendjében és előjelekben lehet eltérés).* □

Az előbbi tétel szerint tehát minden $n \in \mathbb{Z}$, $|n| > 1$ egész szám felírható

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \cdots \cdot p_r^{\alpha_r} \tag{3.1}$$

alakban, ahol a $p_i \in \mathbb{P}$ számok páronként különböző prímszámok, és $\alpha_i \geq 1$. Mivel ez az előállítás (lényegtelen dolgoktól eltekintve) *egyértelmű*, ezért külön nevet is adunk neki:

**3.7. Definíció.** A fenti előállítást az $n$ szám **törzs-** (vagy **prím-**) **tényezős alakjának** (**felbontásának**), vagy $n$ **kanonikus** (rendezett, lat.) **alakjának** hívjuk. □

Ismét hangsúlyozzuk, hogy a legtöbb $n \in \mathbb{N}$ természetes szám (3.1)-ben említett felbontását a gyakorlatban *lehetetlen* (azaz csak évmilliárdok alatt lehetséges) megkeresni – ld. például a 3.2. „*A számelmélet algoritmikus problémái*" alfejezetben található példákat!

**3.8. Megjegyzés.** Mivel a prímszámok tovább már nem osztható számok, ezért **atom**oknak is nevezhetnénk őket (*atom=oszthatatlan*, gör.). A molekulákat atomok építik fel – a természetes számokat prímszámok. A molekulák összegképlete megegyezik a (3.1) képlettel, ezért a (3.1) képletet hívhatnánk az $n \in \mathbb{N}$ szám „*összegképletének*"-nek is.

Pedagógiailag sokszor segített a fenti hasonlat és a következő jelölés, amit a későbbiekben mi is többször használunk:

**3.9. Definíció.** Tetszőleges $n \in \mathbb{N}$, $n > 1$ egész számra $\mathfrak{p}(\mathbf{n})$ jelölje a (3.1) egyenlőségben szereplő prímszámok „multihalmazát" multiplicitással, vagyis $\mathfrak{p}left(n)$-ben a $p_i$ prímszám pontosan $\alpha_i$-szer szerepel:

$$\mathfrak{p}(n) := \{p_1, \ldots, p_1, p_2, \ldots, p_2, \ldots, p_r, \ldots, p_r\} \tag{3.2}$$

(például $\mathfrak{p}(12) = \{2, 2, 3\}$). Legyen továbbá

$$\mathfrak{p}(1) := \varnothing. \quad \square$$

Persze $\mathfrak{p}(n)$ legtöbbször nem halmaz, mert elemei többször is szerepelhetnek benne, de ezzel az (elméleti) problémával most nem foglalkozunk. Negatív $n \in \mathbb{Z}$ számokra is ki *lehetne* terjeszteni a $\mathfrak{p}(n)$ jelölést, de sok probléma merülne fel, feleslegesen ezzel sem foglalkozunk. A $\mathfrak{p}(0)$ halmazt ugyancsak nem definiáljuk.

Könnyen láthatóak az alábbi hasznos tulajdonságok:

**3.10. Állítás.** Tetszőleges $n, m \in \mathbb{N}$, $n, m \geq 1$ *természetes számokra*

$$\mathfrak{p}(n \cdot m) = \mathfrak{p}(n) \cup \mathfrak{p}(m), \tag{3.3}$$

$$n \mid m \quad \Longleftrightarrow \quad \mathfrak{p}(n) \subseteq \mathfrak{p}(m), \tag{3.4}$$

*és* $n \mid m$ *esetén*

$$\mathfrak{p}\left(\frac{m}{n}\right) = \mathfrak{p}(m) \setminus \mathfrak{p}(n). \quad \square \tag{3.5}$$

Még pl. a 3.31. Tétel szerint:

$$lnko(n, m) = \mathfrak{p}(n) \cap \mathfrak{p}(m)$$

és

$$lkkt(n, m) = \mathfrak{p}(n) \cup \mathfrak{p}(m),$$

stb.

A *multihalmazok* közötti műveleteket most ugyan nem definiáltuk, de az Olvasó könnyen kitalálhatja ezeket.

**3.11. Definíció.** Tetszőleges $p \in \mathbb{P}$ prím és $n, \alpha \in \mathbb{Z}$ számokra

$$p^{\alpha} \parallel n \tag{3.6}$$

-t írunk ha $p^{\alpha}$ **pontosan osztja** $n$-t, vagyis:

$$p^{\alpha} \mid n \quad \text{de} \quad p^{\alpha+1} \nmid n. \tag{3.7}$$

**3.12. Definíció.** Az $n \in \mathbb{Z}$ számot **négyzetmentes**nek nevezzük, ha prímfelbontásában minden prímosztója csak egyszer szerepel ($p_i^2 \nmid n$ vagyis $p_i \parallel n$), azaz (3.1)-ben mindegyik $\alpha_i = 1$. $\square$

**3.13. Definíció.** Tetszőleges $n \in \mathbb{Z}^+$ számra jelölje $\mathbf{d}(n)$ az $n$ szám pozitív osztóinak *számát* (**divisors number**). $\square$

**3.14. Állítás.**

$$d(n) = (\alpha_1 + 1) \cdot (\alpha_2 + 1) \cdot \cdots \cdot (\alpha_r + 1)$$

*a (3.1) képlet jelöléseivel.* $\square$

**3.15. Megjegyzés.** **(i)** Ne feledjük, hogy az $n = 0$ számnak *nincs* törzstényezős felbontása, de szerencsére szükségünk sem lesz rá.
**(ii)** Hangsúlyozzuk, hogy a fenti felbontás alapján a legtöbb számelméleti kérdés és vizsgálat egyszerűnek tűnik, de maga a felbontás megkeresése közepesen nagy (egy-kétszáz jegyű) számok esetén *évbilliókig* is eltarthat – még a „modern" több teraHz-es többmagos párhuzamos processzorokkal működő szuperszámítógépekkel is. Ezeket a kérdéseket a következő alfejezetben vizsgáljuk „A Számelmélet Algoritmikus Problémái" címszó alatt.
(iii) Az egész számokhoz hasonlóan más gyűrűkben is vizsgálhatjuk az oszthatóságot, irreducibilis elemeket, prímfelbontást: polinomok, mátriszok, Gauss- és Euler- egészek gyűrűje, stb. (Ezeket a halmazokat a Függelékben ismertetjük röviden.)

## 3.2. A számelmélet algoritmikus problémái

Ebben az alfejezetben a számelméletben felmerülő legtöbb számítási képlet gyakorlati *megvalósíthatatlanságára* hívjuk fel az Olvasó figyelmét.

Bemelegítésképpen próbálja meg az Olvasó felbontani a Bevezetőben 1.1. Példában felírt számokat (a megoldás a jelen fejezet végén a 3.25. Megoldásban található).

Igen, a bajok már a 7-8 jegyű számokkal elkezdődnek, pedig a modern alkalmazásokban többszáz, -ezer jegyű egész- és prímszámokkal kellene számolnunk. Nem az összeadással és szorzással van baj – ezeket többszázjegyű számokkal is pillanatok alatt elvégzi *bárki*, hanem hogy sok ilyen osztást kell elvégeznünk.

**3.16. Probléma.** ***A Számelmélet Algoritmikus Problémái.***
*Legyen az input egy* tetszőleges *(többszáz/ezer-jegyű)* $n \in \mathbb{N}$ *természetes szám.*
***(i) Prímtesztelés probléma****:* $n$ *prímszám-e?*
***(ii) Prímfelbontás (faktorizációs) probléma****:* *ha* $n$ *nem prímszám, akkor bontsuk fel legalább két (nála kisebb) szám szorzatára!*
***(iii) Prímgenerálás probléma****:* *adjunk meg (legalább egy)* $n$*-nél nagyobb* $p$ *prímszámot!*
□

Nagyon jól gondoljuk meg a három probléma különbözőségét!

**3.17. Megjegyzés.** Nyilván a **(ii)** probléma megoldásával az **(i)** problémára is adnánk megoldást, sőt a **(iii)** probléma megoldásához is közelebb kerülnénk.

**3.18. Megjegyzés.** Mindhárom problémára **van** algoritmus, amely *véges idő alatt* ad megoldást: az Eratoszthéneszi szita-algoritmus.

**3.19. Algoritmus. Eratoszthéneszi szita.** Adott *tetszőleges* $n \in \mathbb{N}$ szám prímtényezős felbontására.

Az algoritmus jól ismert: az $n$ számot elosztjuk a $\sqrt{n}$-nél kisebb páratlan számokkal.
□

**3.20. Megjegyzés.** **(i)** Eratoszthenesz (Kr.e.276–196) görög matematikus.
**(ii)** A fenti 3.19. Algoritmus exponenciálisan lassú.
Ezt a 2.10. Példában már megállapítottuk a 2.2.1. „Természetes számok mérete” alfejezetben.
□

**3.21. Példa.** **(a)** Mennyi ideig fut az Eratoszthéneszi szita-algoritmus egy $k$ **-jegyű** input esetén, a $k = 20$, $k = 30$, $k = 40$ és $k = 50$, $k = 100$, ... esetekben egy *5GHz*-es gépen futtatva (ha csak az osztásokat számítjuk egy-egy lépésnek, azaz feltételezzük, hogy a gép minden órajel alatt elvégez egy $k$ jegyű osztást (!) és ellenőrzést, vagyis másodpercenként $5 \cdot 10^9$ osztást)?
**(b)** Mennyire csökkenne a futásidő, ha a $\sqrt{n}$ alatti prímszámokat egy tömbben (táblázatban) tárolnánk, és csak e prímszámokat próbálnánk ki?
**(c)** Mi változna, ha mondjuk 1000-szer gyorsabb gépünk lenne?

**Megoldások** (A részletes számítások megtalálhatók [SzI0]-ban.)

**(a)** Az osztások száma $\frac{\sqrt{n}}{2} \approx 10^{k/2}$, ez

k=20 esetén $5{\cdot}10^9$ lépés = 1 mp,
k=30 esetén $5{\cdot}10^{14}$ lépés = $10^5$ mp $\approx$ 27 óra 46 perc,
k=40 esetén $5{\cdot}10^{19}$ lépés = $10^{10}$ mp $\approx$ 317 év 35 nap 18 óra,
k=50 esetén $5{\cdot}10^{24}$ lépés = $10^{15}$ mp $\approx$ 31,7 millió év,
k=100 esetén $5{\cdot}10^{49}$ lépés = $10^{40}$ mp $\approx 3,17 \times 10^{23}$ *milliárd év* ....

**(b)** Ha $\pi(x)$ jelöli az $x \in \mathbb{R}$-nél kisebb prímszámok számát (ld. 3.50. Definíció), akkor a 3.51. „Nagy Prímszámtétel" szerint $\pi(x) \sim \frac{x}{\ln(x)}$, vagyis az osztások száma $\pi(\sqrt{n}) \sim \frac{\sqrt{n}}{\ln(\sqrt{n})}$. (Az $f \sim g$ jelölést a 2.5. Definícióban vezettük be.)

Az időadatok:

k=20 esetén $\approx 4{,}3{\cdot}10^8$ lépés $<$ 1 mp,
k=30 esetén $\approx 2{,}9{\cdot}10^{13}$ lépés $\approx$ 5790 mp $\approx$ 1 óra 36 perc,
k=40 esetén $\approx 2{,}2{\cdot}10^{18}$ lépés $\approx 4{,}4{\cdot}10^8$ mp $\approx$ 13 év 281 nap 13 óra,
k=50 esetén $\approx 1{,}4{\cdot}10^{23}$ lépés $\approx 1{,}7{\cdot}10^{14}$ mp $\approx$ 5,5 millió év ....
k=100 esetén HF.

**(c)** Semmi. □

A könyvünkhöz mellékelt PRIM1D.EXE program éppen az Eratoszthenészi szitamódszer *lassúságát* szemlélteti: hűségesen végigpróbálgatja az összes, $\sqrt{n}$-nél kisebb páratlan számot. Könyvünk egyik célja éppen a lehetséges gyorsítások bemutatása (ld. a 7. "*Prímtesztelés és számok felbontása*" fejezetben).

Sok esetben a prímek beolvasása fájlból még lassíthatja is a program futását összességében, erre a 11. "*Számítógépes megvalósítások*" fejezetben találunk példákat. Például, az egymilliomodik prímszám $p_{1\ 000\ 000} = 15\ 485\ 863$ még csak 8 jegyű (és ráadásul 1-gyel kezdődik), azaz $\pi(15\ 485\ 863) = 1\ 000\ 000$.

Könyvünk fő célja a fenti 3.16. Problémára gyorsabb algoritmusokat mutatni, bár csak néhány egyszerűbbre van helyünk. (Lásd a 7. "*Prímtesztelés és számok felbontása*" és a 8. "*Prímkeresés*" fejezeteket, „természetesen" előtte sok elméletet kell megismernünk: ld. 3-6. fejezetek.) Bonyolultabb (jobb) algoritmusokat például Knuth [KD] nagyszerű művében találhatunk.

## A legfontosabb eredmények

**3.22. Állítás.** *A 3.16.**(ii)** (Prímfelbontás) problémára nem ismert gyors (polinomiális) algoritmus.* □

Tehát óvatosan fogadjunk minden olyan tételt és képletet, amely használja a prímfelbontás (3.1) képletet! Megjegyezzük: éppen ez a jó, mert ellenkező esetben a ma használatos titkosírások könnyen feltörhetőek lennének!

Még egyszer hangsúlyozzuk: érdekes módon a nagy (egész) számokkal való műveletek a modern számítógépek robbanásszerű fejlődése és tömeges elterjedése, valamint a számelmélet többezer éves eredményei ellenére megközelíthetetlen problémákat támasztanak, de éppen

ezért váltak fontossá a kódolás, jelek zajszűrése, titkosírások, aláírás hitelesítése, jelszóvédelem problémák megoldásához.

Vannak azonban olyan problémák, melyeket meg tudunk oldani prímfelbontás *nélkül*, mint például *Euklidesz* (Kr.e.300 !!!) eredménye $lnko$ kiszámítására és elsőfokú Diophantoszi egyenletek megoldására (ld.a 4.2. "*Euklidesz algoritmusa*" alfejezetet és a 4. "*Lineáris Diophantoszi egyenletek*" fejezetet).

**3.23. Tétel** (Agrawal–Kayal–Saxena, 2001 [AKS])**.** *A 3.16.**(i)** (Prímtesztelés) problémára **van** gyors (polinomiális) algoritmus.*
*Az algoritmus hivatalos rövidítése: **AKS**-algoritmus.* □

**3.24. Megjegyzés.** A 3.16.**(iii)** (Prímgenerálás) problémára már több mint száz éve léteznek nem túl lassú (bár nem is túl gyors) algoritmusok, ezeket a 8. „Prímkeresés" fejezetben tárgyaljuk.
Az Érdeklődők figyelmét felhívjuk az Interneten zajló nagy prímkeresési programra (többezer $ jutalommal!): HTTP://WWW.MERSENNE.ORG és HTTP://PRIMES.UTM.EDU □

## Megoldások

**3.25. Megoldás. Az 1.1. Példában említett számok felbontásai:**
**a)** $440\,747 = 613 \cdot 719$,
**b)** $2\,347\,589 = 1483 \cdot 1583$,
**c)** $97\,189\,241 = 7151 \cdot 13\,591$,
**d)** $17967876255379 = 81371 \cdot 220814249$,
**e)** $444113096135661846937 = 3719977867 \cdot 119385951211$,
**f)** $2^{67} - 1 = 193707721 \cdot 761838257287$
**g)** A megoldás történetét (600 számítógépen több mint 8 hónap 1994-ben) a 9. "*Titkosírás nyilvános kulccsal*" fejezetben a 10.17. Feladatnál, míg a végeredményt a 10.23. Megoldásnál ismertetjük. □

# 3.3. lnko és lkkt

**3.26. Definíció.** Azt mondjuk, hogy $d$ **közös osztója** az $a$ és $b$ egész számoknak, ha $d|a$ és $d|b$.
Azt mondjuk, hogy $g$ a **legnagyobb közös osztója** $a$-nak és $b$-nek, ha $g$ a közös osztók közül a legnagyobb, azaz, ha $d$ is közös osztója $a$-nak és $b$-nek, akkor $d \leq g$. Ennek jelölése $\mathbf{lnko(a,b)}$ vagy $\mathbf{gcd(a,b)}$ (greatest common divisor) vagy csak röviden $(\mathbf{a},\mathbf{b})$.
Hasonlóan az $a_1, \ldots, a_n \in \mathbb{Z}$ számok közös osztói közül a legnagyobbat, azaz a számok legnagyobb közös osztóját $\mathbf{lnko}(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ vagy csak $(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ jelöli. □

**3.27. Megjegyzés.** Bármely két egész számnak 1 és $-1$ is közös osztója. Mivel egy *(nem nulla)* egész számnak véges sok osztója van, ezért közös osztókból is csak véges sok van, ezért a legnagyobb közös osztó mindig egyértelműen definiált, *pozitív*, sőt $1 \leq lnko(a,b) \leq \min\{|a|,|b|\}$ bármilyen $a, b \in \mathbb{Z}$ (akár negatív akár pozitív) számokra. □

**3.28. Definíció.** A $h \in \mathbb{Z}$ egész számot az $a$ és $b$ egész számok **közös többszörösé**nek nevezzük, ha $a|h$ és $b|h$. Az $a$ és $b$ számok közös *pozitív* többszörösei közül a legkisebbet az $a$ és $b$ **legkisebb közös többszörösé**nek hívjuk, és $\mathbf{lkkt}(\mathbf{a}, \mathbf{b})$ vagy $\mathrm{lcm}\,(\mathbf{a}, \mathbf{b})$ (least common multiplier) vagy röviden $[\mathbf{a}, \mathbf{b}]$-vel jelöljük.
Hasonlóan, az $a_1, \ldots, a_n \in \mathbb{Z}$ számok közös pozitív többszörösei közül a legkisebbet, azaz a számok legkisebb közös többszörösét $\mathbf{lkkt}(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ vagy csak röviden $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$ jelöli. □

**3.29. Megjegyzés.** Bár széles körben elterjedtek az $(a, b)$ és $[a, b]$ jelölések, mi *kizárólag* csak az $lnko(a, b)$, $lkkt(a, b)$ jelöléseket használjuk az esetleges félreértések elkerülése végett. □

**3.30. Tétel.** *Legyen* $a = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ *és* $b = p_1^{\beta_1} p_2^{\beta_2} \cdots p_r^{\beta_r}$ *ahol* $0 \leq \alpha_i,\ \beta_i$. *Ekkor*

$$\begin{aligned} lnko(a, b) &= p_1^{\min(\alpha_1, \beta_1)} \cdot p_2^{\min(\alpha_2, \beta_2)} \cdot \ldots \cdot p_r^{\min(\alpha_r, \beta_r)}, \\ lkkt\,(a, b) &= p_1^{\max(\alpha_1, \beta_1)} \cdot p_2^{\max(\alpha_2, \beta_2)} \cdot \ldots \cdot p_r^{\max(\alpha_r, \beta_r)}. \end{aligned} \tag{3.8}$$

*Hasonlóan több számra:*

$$\begin{aligned} lnko(a, b, \ldots, z) &= p_1^{\min(\alpha_1, \ldots, \omega_1)} \cdot p_2^{\min(\alpha_2, \ldots, \omega_2)} \cdot \ldots \cdot p_r^{\min(\alpha_r, \ldots, \omega_r)} \\ lkkt\,(a, b, \ldots, z) &= p_1^{\max(\alpha_1, \ldots, \omega_1)} \cdot p_2^{\max(\alpha_2, \ldots, \omega_2)} \cdot \ldots \cdot p_r^{\max(\alpha_r, \ldots, \omega_r)} \end{aligned}$$

*ahol* $z = p_1^{\omega_1} p_2^{\omega_2} \cdots p_r^{\omega_r}$ *ahol* $0 \leq \omega_i$. □

Az előző eredmény szemléletes változata a $\mathfrak{p}\,(n)$ jelöléssel (ld.(3.9)):

**3.31. Tétel.** *Tetszőleges* $a_1, \ldots, a_t \in \mathbb{Z}$ *számokra*

$$\begin{aligned} lnko(a_1, \ldots, a_t) &= \mathfrak{p}(a_1) \cap \cdots \cap\ \mathfrak{p}(a_t), \\ lkkt(a_1, \ldots, a_t) &= \mathfrak{p}(a_1) \cup \cdots \cup\ \mathfrak{p}(a_t). \end{aligned}$$ □

**3.32. Megjegyzés.** A 3.2. "*A számelmélet algoritmikus problémái*" alfejezetben említett algoritmikus problémák miatt a fenti képletek csak *elméleti* jelentőségűek. Felhívjuk a figyelmet, hogy $lnko$ és $lkkt$ értékét a gyakorlatban mégis *gyorsan* ki lehet számítani egy másik módszerrel: *Euklidesz algoritmusával*, amit (további alkalmazásokkal) a 4.2. „Euklidesz algoritmusa" alfejezetben ismertetünk.
Az Euklideszi algoritmus előnye *nem csak* a gyorsasága! Ha vele a legnagyobb közös osztót az argumentumok prímtényezős felbontása nélkül is ki lehet számolni, akkor prímtesztelő és -felbontó algoritmusoknál a vizsgálandó („prím vagy összetett?"), ismeretlen számok lnkoját is ki tudjuk számolni! Erre pedig számtalanszor lesz szükségünk a 8. „*Prímtesztelés és számok felbontása*" fejezetben!

A $\mathfrak{p}\,(a)$ és az alábbi $\Delta$, $\nabla$ jelölésekkel azonban sok hasznos összefüggést talán könyebben megérthetünk:

**3.33. Definíció.** Tetszőleges $a, b \in \mathbb{Z}$ számok esetén legyen

$$\mathbf{a\Delta b} := lnko(a,b),$$
$$\mathbf{a\nabla b} := lkkt(a,b). \quad \square$$

Ugyanis: $\Delta$ és $\nabla$ nem csak kommutatív műveletek:

**3.34. Állítás.** ***(kommutativitás)***

$$a\Delta b = b\Delta a \quad azaz \quad lnko(a,b) = lnko(b,a)\,,$$
$$a\nabla b = b\nabla a \quad azaz \quad lkkt(a,b) = lkkt(b,a), \quad \square$$

hanem asszociatívok

**3.35. Tétel** (asszociativitás)**.** *Tetszőleges $a, b, c \in \mathbb{Z}$ számokra*

$$(a\Delta b)\,\Delta c = a\Delta\,(b\Delta c)\,,$$
$$[a\nabla b]\,\nabla c = a\nabla\,[b\nabla c]\,,$$

*hiszen jól ismert:*

$$lnko\,(lnko(a,b),\,c) = lnko\,(a,\,lnko(b,c))\,,$$
$$lkkt\,(lkkt\,(a,b)\,,\,c) = lkkt\,(a,\,lkkt\,(b,c))\,.$$

és disztributívok is:

**3.36. Tétel** (disztributivitás)**.** *Tetszőleges $a, b, c \in \mathbb{Z}$ számokra*

$$(a\Delta b)\,\nabla c = [a\nabla c]\,\Delta\,[b\nabla c]\,, \tag{3.9}$$
$$[a\nabla b]\,\Delta c = (a\Delta c)\,\nabla\,(b\Delta c)\,,$$

*azaz*

$$lkkt\,(lnko\,(a,b)\,,c) = lnko\,(lkkt\,(a,c)\,,lkkt\,(b,c))\,, \tag{3.10}$$
$$lnko\,(lkkt\,(a,b)\,,c) = lkkt\,(lnko\,(a,c)\,,lnko\,(b,c))\,.$$

A fenti összefüggések nem meglepők a 3.30. Tétel (3.8) klasszikus összefüggései, vagy az alábbi szemléletes bizonyítás alapján:

**Bizonyítás.**

$$(\mathfrak{p}\,(a) \cap \mathfrak{p}\,(b)) \;\cap \mathfrak{p}\,(c) = \mathfrak{p}\,(a) \cap\; (\mathfrak{p}\,(b) \cap \mathfrak{p}\,(c)) = \mathfrak{p}\,(a) \cap \mathfrak{p}\,(b) \cap \mathfrak{p}\,(c)\,, \tag{3.11}$$
$$(\mathfrak{p}\,(a) \cup \mathfrak{p}\,(b)) \;\cup \mathfrak{p}\,(c) = \mathfrak{p}\,(a) \cup\; (\mathfrak{p}\,(b) \cup \mathfrak{p}\,(c)) = \mathfrak{p}\,(a) \cup \mathfrak{p}\,(b) \cup \mathfrak{p}\,(c)\,,$$

és

$$(\mathfrak{p}\,(a) \cap \mathfrak{p}\,(b)) \;\cup \mathfrak{p}\,(c) = (\mathfrak{p}\,(a) \cup \mathfrak{p}\,(c)) \;\cap\; (\mathfrak{p}\,(b) \cup \mathfrak{p}\,(c)) \tag{3.12}$$
$$(\mathfrak{p}\,(a) \cup \mathfrak{p}\,(b)) \;\cap \mathfrak{p}\,(c) = (\mathfrak{p}\,(a) \cap \mathfrak{p}\,(c)) \;\cup\; (\mathfrak{p}\,(b) \cap \mathfrak{p}\,(c))$$

■

és hasonlóan:

**3.37. Tétel** (több szám)**.** *Tetszőleges $a, b, c \in \mathbb{Z}$ számokra*

$$lnko(a,b,c) = lnko\left(lnko(a,b), c\right),$$
$$lkkt(a,b,c) = lkkt\left(lkkt(a,b), c\right).$$

□

A fenti Tétel szerint *akárhány* szám legnagyobb közös osztóját illetve legkisebb közös többszörösét vissza lehet vezetni két szám legnagyobb közös osztójának illetve legkisebb közös többszörösének kiszámítására, ami a gyakorlati alkalmazásoknál felbecsülhetetlen segítség (ld. 4.2. "*Euklidesz algoritmusa*" fejezetben).

A $\mathfrak{p}\,(a)$ jelölés segítségével érthetjük meg $lnko$ és $lkkt$ ($\Delta$ és $\nabla$) többi tulajdonságait is, amit legrövidebben az alábbi Tételben foglalhatunk össze:

**3.38. Tétel.** *Legyen $n \in \mathbb{N}$ egy tetszőleges négyzetmentes szám (ld.3.12. Definíció). Ekkor a*

$$\left(D_n,\, lnko,\, lkkt,\, \frac{n}{x},\, n,\, 1\right)$$

*hatos (struktúra)* ***Boole algebra****, azaz teljesíti a halmazműveletek:*

$$\left(\,\mathcal{P}\,(H)\,,\, \cap,\, \cup,\, \overline{A},\, H,\, \varnothing\right)$$

*jólismert tulajdonságait (pl. Függelék (BA1)-(BA14) ),*
*ahol*

$$D_n := \{\ n \text{ osztóinak halmaza } \}$$

*és $\mathcal{P}\,(H)$ a $H$ halmaz hatványhalmaza.* □

**3.39. Példa.** Például a jól ismert $\overline{A \cup B} = \overline{A} \cap \overline{B}$ DeMorgan azonosság a számelmélet nyelvén:

$$\frac{n}{lkkt\,(x,y)} = lnko\left(\frac{n}{x}, \frac{n}{y}\right). \qquad \square$$

A számelmélet egyik legfontosabb fogalma a következő:

**3.40. Definíció.** Azt mondjuk, hogy $a$ és $b$ **relatív prímek** ($a$ **is prime to** $b$, $a$ **is coprime to** $b$), ha $\quad lnko(a,b) = 1.$ □

Másképpen fogalmazva:

**3.41. Állítás.** *Tetszőleges $a, b \in \mathbb{Z}$ számok pontosan akkor relatív prímek, ha nincs közös osztójuk, vagyis bármely $u, v \in \mathbb{Z}$ számokra ($u \neq 1$, $v \neq 1$)*

$$u \mid a \Longrightarrow u \nmid b \quad \textit{és} \quad v \mid b \Longrightarrow v \nmid a,$$

*szemléletesen:*

$$\mathfrak{p}\,(a) \cap \mathfrak{p}\,(b) = \emptyset. \qquad \square$$

A 3.40. Definícióban említett angol elnevezés nem pontos, mert ez egy szimmetrikus reláció, hiszen $lnko$ kommutatív művelet (vagyis $lnko\,(a,b)=1$ pontosan akkor, ha $lnko\,(b,a)=1$).

A *relatív prím* elnevezés tehát azt fejezi ki, hogy $a$ „szemszögéből" $b$ prím, vagyis ha *csak* $a$ osztóit tekintjük, akkor $b$ ezek egyikével sem osztható vagyis $b$ prím $a$ (osztói)-hoz viszonyítva, és megfordítva is.

A fenti 3.41. Állítás alapján könnyen belátható a következő hasznos összefüggés is:

**3.42. Állítás.** *$a$ pontosan akkor relatív prím $b$-hez, ha $b$ összes osztójához is relatív prím.*

*Másként fogalmazva: $a$ pontosan akkor relatív prím az $m$ és $n$ számokhoz, ha relatív prím $m\cdot n$-hez.* □

Több szám relatív prím kapcsolatát többféleképpen is lehet értelmezni, azonban ezek a definíciók között nagy eltérés van – erre ügyelnünk kell például a 7. "*Kínai maradéktétel ...*" fejezetben.

**3.43. Definíció.** **(i)** az $a_1,\dots,a_t\in\mathbb{Z}$ tetszőleges számok **relatív prímek**, ha

$$lnko\,(a_1,\dots,a_t)=1, \tag{3.13}$$

**(ii)** az $a_1,\dots,a_t\in\mathbb{Z}$ tetszőleges számok **páronként relatív prímek**, ha

$$lnko\,(a_i,a_j)=1 \quad \text{minden } i\neq j \text{ párra.} \tag{3.14}$$

□

Felhívjuk a figyelmet, hogy (ii) sokkal erősebb követelmény (i)-nél!

Néhány további közismert és hasznos összefüggés:

**3.44. Tétel.** *Tetszőleges $a,b\in\mathbb{Z}$ esetén*

$$lkkt(a,b)=\frac{a\cdot b}{lnko(a,b)}. \qquad \square$$

**3.45. Tétel.** ***(i)*** *minden $m\in\mathbb{Z}$ egészre*

$$lnko(m\cdot a,\ m\cdot b)=|m|\cdot lnko(a,b),$$

***(ii)*** *az $a$ és $b$ egész számok bármely $d$ közös osztójára*

$$lnko\left(\frac{a}{d},\frac{b}{d}\right)=\frac{1}{d}\cdot lnko(a,b),$$

***(iii)*** $lnko\left(\dfrac{a}{g},\dfrac{b}{g}\right)=1$ *akkor és csak akkor, ha* $g=lnko(a,b)$,

*vagyis $\dfrac{a}{d}$ és $\dfrac{b}{d}$ relatív prímek ahol $d=lnko\,(a,b)$,*

***(iv)*** *ha $(a,c)=1$ és $(b,c)=1$, akkor $(ab,c)=1$,*

***(v)*** *ha $c|ab$ és $(b,c)=1$, akkor $c|a$.* □

Az *Euklideszi Algoritmus* (3.2.alfejezet) segítségével könnyen igazolható az alábbi, sokszor hasznos összefüggés:

**3.46. Állítás.** *Tetszőleges $a, m, n \in \mathbb{N}$ természetes számokra*

$$lnko\,(a^m - 1,\, a^n - 1) = a^{lnko(m,n)} - 1. \quad \square$$

A 9.1. "*Az RSA-algoritmus*" alfejezetben szükségünk lesz a következő eredményre:

**3.47. Tétel** (Dirichlet, 1849)**.** *Tetszőlegesen választott $u, v \in \mathbb{N}$ természetes számok $6/\pi^2 \approx 0,60793$ valószínűséggel lesznek relatív prímek.* $\square$

## 3.4. A prímszámok eloszlása

Bár a prímszámokkal kapcsolatban még mindig nagyon sok kérdésre nem tudunk válaszolni, körülbelüli eloszlásukról elég sokat tudunk.

Kezdjük két elemi tétellel.

**3.48. Tétel** (Euklidesz)**.** *A prímszámok száma végtelen.*

**Bizonyítás.** Ha $p_1, p_2, \ldots, p_k$ prímszámok, akkor az

$$N := p_1 \cdot p_2 \cdot \cdots \cdot p_k + 1$$

szám egyik előző $p_i$ számmal sem osztható, tehát vagy maga egy (új) prím, vagy az (összes) osztója új, az előzőektől különböző prímszám. ■

**3.49. Tétel.** *A prímszámok (növekvő) sorozatában két prímszám közötti távolság tetszőlegesen nagy lehet, azaz bármely $\ell$* pozitív *egész számhoz létezik $\ell$ db egymás utáni összetett szám.*

**Bizonyítás.** Tetszőleges $k \in \mathbb{N}$, $k \geq 2$ szám esetén a $k! + 2, k! + 3, \ldots, k! + k$ számok egyike sem prím, hiszen rendre 2-vel, 3-mal, …, $k$-val osztható, ez $\ell = k - 1$ db szám, és $k$ tetszőlegesen nagy lehet. ■

A prímszámok és számtani sorozatok kapcsolatával a "*Számtani sorozatok*" valamint az "*Ikerprímek*" alfejezetekben foglalkozunk.

Pontosabb eredményeket ismerünk a prímszámok eloszlásáról, ezek bizonyítása azonban jóval nehezebb:

**3.50. Definíció. (i)** Tetszőleges $n \in \mathbb{N}$ természetes szám esetén jelölje $\mathbf{p_n}$ az $n$ **-edik** *(pozitív)* **prímszámot**,
**(ii)** tetszőleges $x \in \mathbb{R}^+$ szám esetén jelölje $\pi(\mathbf{x})$ az $x$-nél kisebb *(pozitív)* **prímszámok számát**. $\square$

Érdekességképpen: pl. az egymilliomodik prímszám még csak $p_{1\ 000\ 000} = 15\ 485\ 863$ azaz $\pi\,(15\ 485\ 863) = 1\ 000\ 000$. Ugye milyen sok nyolcjegyű prímszám van!?

**3.51. Tétel** (Nagy Prímszámtétel)**.**

$$\lim_{n\to\infty} \frac{\pi(n)}{\frac{n}{\log n}} = 1,$$

*azaz: nagy $n$-re*

$$\pi(n) \sim \frac{n}{\log n},$$

*más formában:*

$$\pi(n) \sim \int_2^n \frac{1}{\ln(x)}\, dx,$$

*pontosabb becslés: bármely $r \in \mathbb{N}$ természetes számra*

$$\pi(n) = \frac{n}{\log(n)} + \frac{1!\cdot n}{\log^2(n)} + \frac{2!\cdot n}{\log^3(n)} + \cdots + \frac{r!\cdot n}{\log^{r+1}(n)} + \mathcal{O}\left(\frac{n}{\log^{r+2}(n)}\right).$$

□

(Az $f \sim g$ jelölést a 2.5. Definícióban vezettük be.)

A legelső összefüggést *Carl Friedrich Gauss (1777–1855) német matematikus* sejtette meg, bizonyítani azonban csak *Jacques Hadamard (1865-1963)* és *Charles Jean de la Vallée Poussin (1866-1962) francia matematikusok*nak sikerült 1896-ban.

**3.52. Tétel** (Csebisev)**.** *Tetszőleges $n \in \mathbb{N}$, $n > 1$ számra $n$ és $2n$ között mindig van prímszám.* □

*(Pafnutyij Lvovics Csebisev (1821-1894) orosz matematikus.)*

A sok tétel közül most csak kettőt említünk meg:

**3.53. Tétel.**

$$p_n \sim n \cdot \log(n). \quad \square$$

**3.54. Tétel.** *Ha $x \in \mathbb{R}$, $2 \le x$, akkor*

$$\sum_{\substack{p\le x\\ p\in\mathbb{P}}} \frac{1}{p} = \log(\log(x)) + o(1)$$

*azaz*

$$\lim_{x\to\infty} \left( \sum_{\substack{p\le x\\ p\in\mathbb{P}}} \frac{1}{p} - \log(\log(x)) \right) = 0. \quad \square$$

## 3.5. Nevezetes problémák

Most csak néhány számunkra érdekes problémát „villantunk fel”, amelyek jól mutatják az elméleti és a számítási nehézségeket.

### 3.5.1. Pitagorasz és FLT

**3.55. Definíció.** Az

$$x^2 + y^2 = z^2 \tag{3.15}$$

egyenlet pozitív egész megoldásait **Pitagoraszi számhármasok**nak (**Pithagorean triplets**) nevezzük. □

(*Pitagorasz Kr.e. VI. században élt görög matematikus.*)
Már Pitagorasz maga és az ókori Babilóniaiak is ismerték az alábbi képletet:

**3.56. Állítás.** *A (3.15) egyenlet összes megoldása:* $x = 2m$, $y = m^2 - 1$, $z = m^2 + 1$ *ahol* $m \in \mathbb{N}$ *tetszőleges természetes szám.* □

Pierre Fermat az 1600-as évek közepén egy könyv margójára a következő állítást írta (az állítást fia hozta nyilvánosságra 1670-ben):

**3.57. Állítás.** ***(Fermat Nagy Sejtése)*** *„A (3.15) egyenletnek azonban* $n \geq 3$ *esetén nincsen pozitív egész gyöke. Erre egy csodálatos bizonyítást találtam, de a lap széle túl keskeny ahhoz, hogy azt befogadja."* □

A fenti sejtést angolul **Fermat's Last Theorem** (Fermat utolsó tétele) vagy csak röviden **FLT**-nek hívják.

Máig rejtély, hogy mi volt Fermat „csodálatos" bizonyítása.
Néhány legelső bizonyítás ($n$ speciális értékeire) a $Z[\alpha]$ halmazok segítségével sikerült, amely halmazokat és alaptulajdonságaikat a *Függelékben* ismertetjük.
A bizonyítás hiányzó láncszemét 1993-ban *Andrew Wiles*nak sikerült pótolnia, amit végleges formában 1995-ben publikált *R. Taylor*ral közösen.

### 3.5.2. Karácsonyi Tétel és Bolyai János

**3.58. Tétel** (Fermat karácsonyi tétele)**.** *Minden* $4m+1$ *alakú prímszám előáll két négyzetszám összegeként.* □

Bolyai János nagyon egyszerű bizonyítást talált a fenti tételre a $Z[\alpha]$ halmazok segítségével (ld. *Függelékben*). Erről így ír Kiss Elemér [KE1] és [KE2]-ben:

"*Ugyancsak Bolyai Farkas bíztatta fiát arra, hogy keresse meg a fent már említett Fermat karácsonyi tételének „legegyszerűbb" bizonyítását. A tételt Fermat fogalmazta meg 1640 karácsonyán, de csak jóval később L. Euler bizonyította be egy* hosszú, 55 oldalas *dolgozatban. Bolyai János – felhasználva a komplex egészek elméletét –* ***négy bizonyítást*** *is talált a tételre. Ezek közül az egyik különösen rövid és egyszerű.* Mindössze két sor. *A XX. század matematikusai valósággal versenyeztek azon, hogy ki tudna minél egyszerűbb bizonyítást találni Fermat-tételére. Ezek a kísérletek Don Zagier 1990-es dolgozatában csúcsosodtak ki, amelyben ő „egy mondatban" bizonyította be a tételt. Sokan úgy gondolják, hogy ez a tételre adható legszebb bizonyítás, következésképpen ez „került be" az Erdős Pál által oly sokszor emlegetett* Nagy Könyv*be. Véleményünk szerint a Nagy Könyvben nem Zagier, hanem Bolyai János bizonyítása található."*

### 3.5.3. Számtani sorozatok

**3.59. Tétel** (Dirichlet, 1837)**.** *Ha egy számtani sorozat első tagja és különbsége* relatív prímek*, akkor ebben a számtani sorozatban végtelen sok prímszám van.* □

*(Peter Gustav Lejeune Dirichlet (1805-1859) német matematikus.)*

Dirichlet fenti tételének bizonyítása megtalálható [SzM]-ben vagy a HTTP://WWW.TYPOTEX.HU/?PAGE=KONYVEK&ISBN=978-9-639132-25-2 címen.

**3.60. Probléma.** *Milyen hosszú olyan számtani sorozat van, amelynek minden tagja prímszám?* □

A *2005. novemberében* ismert leghosszabb sorozatok csak 22 tagúak, például:

$$a_n = 28\,383\,220\,937\,263 + 1\,861\,263\,814\,410 \cdot k \qquad (0 \le k \le 21)$$

ahol

$$1\,861\,263\,814\,410 = 2 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 103$$

**3.61. Tétel** (Ben Green, Terence Tao, 2004)**.** *Létezik tetszőleges véges hosszúságú, csak prímekből álló számtani sorozat.* □

(Ld. [FR] és HTTP://ARXIV.ORG/ABS/MATH/0404188, HTTP://FRONT.MATH.UCDAVIS.EDU/MATH.NT/0404188 vagy HTTP://MATHWORLD.WOLFRAM.COM/PRIMEARITHMETICPROGRESSION.HTML.)

Nem tudjuk, hogy **miképpen** lehet egy 23 tagú vagy hosszabb ilyen sorozatot találni.

### 3.5.4. Ikerprímek

Máig nem sikerült bizonyítani az (ókori keltezésű) ún. ikerprím-problémát:

**3.62. Definíció.** A $p, q \in \mathbb{P}$ prímszámokat **ikerprímek**nek nevezzük, ha $q = p + 2$. □

**3.63. Sejtés.** ***(Ikerprím-probléma)*** *Végtelen sok* $(p, q)$ *ikerprím-pár létezik.* □

Az eddigi legnagyobb ikerprímeket *Járai Antal* és munkatársai találták meg 2005 szeptember 9-én. Ezek a prímek:

$$d_{1,2} = 16\,869\,987\,339\,975 \cdot 2^{171\,960} \pm 1 \qquad ([\text{FR}]).$$

(Hány jegyűek is?)

# 4. fejezet

# Maradékos osztás és Euklidesz algoritmusa

A 3.2. "*A számelmélet algoritmikus problémái*" alfejezetben láttuk, hogy gyakorlatilag lehetetlen („nagyméretű") természetes számokat prímtényezőkre bontani – és ez az összes számelméleti probléma gyökere. Azonban $lnko\,(a,b)$ kiszámítására létezik egy meglepően *egyszerű* és *gyors* algoritmus, amit ráadásul már több mint *kétezer* éve felfedeztek és használtak (és még manapság is a legjobb).

Ne feledjük, hogy a [SzI1] Feladatgyűjteményben nagyon sok, részletesen kidolgozott példa található ehhez a fejezethez is.

## 4.1. Maradékos osztás

Ha csak egész számokkal dolgozunk, akkor osztáskor természetes, hogy (majdnem) mindig marad valami a „végén". Ez az egyszerű tény a számelmélet egyik alapfogalma:

**4.1. Tétel** (Maradékos osztás tétele)**.** *Minden $a \in \mathbb{Z}$ és $b > 0$ egész számokhoz létezik olyan,* egyértelműen *meghatározott $q$ és $r$ egész szám, amelyre*

$$a = b \cdot q + r \quad \textbf{\textit{és}} \quad 0 \leq r < b. \quad \square \tag{4.1}$$

Meglepő módon a későbbiekben nem a hányados, hanem a **maradék** lesz a lényeges. Számológépeken ugyan nincs ilyen gomb, de egyes programnyelvekben használatos az

$$r \;:=\; (a \bmod b)$$

vagy hasonló jelölés.

A fenti tételt kiterjeszthetnénk tetszőleges $b \in \mathbb{Z}$ egész számokra is: (4.1) helyett

$$a = b \cdot q + r \quad \textbf{és} \quad 0 \leq r < |b| \tag{4.2}$$

-t kellene írnunk, de ez a gyakorlatban csak felesleges bonyolítás lenne.

**4.2. Megjegyzés.** A valós együtthatójú polinomok $R[x]$ halmazában, a komplex számok

$$\mathbb{Z}[i] := \{a + bi : a, b \in \mathbb{Z}\}$$

„Gauss-egészek”, a

$$\mathbb{Z}[\rho] := \{a + b\rho : a, b \in \mathbb{Z}\}$$

„Euler-egészek” és még sok $Z[\alpha]$ részhalmazában ($\rho = \frac{-1}{2} + \frac{\sqrt{3}}{2}i$, $\alpha \in C$ megfelelő másodfokú algebrai szám) is érvényes a maradékos osztás (4.2) megfelelője (pl. „polinomosztás”). Mivel pedig a jelen fejezet és az utána következő három fejezet a (4.2) egyenlőségre épül, így nem meglepő, hogy ezekben a fejezetekben megismert módszerek (Diophantoszi egyenletek, Kínai maradéktétel, stb.) mind érvényesek a polinomok, Gauss-, Euler egészek és a megfelelő $Z[\alpha]$ részhalmazokban is. Sőt, a bemutatottaknál még általánosabb Euklideszi gyűrűkben is érvényes a maradékos osztás és annak minden következménye, ezt röviden a Függelékben ismertetjük. (A polinomok maradékos osztása a polinomok egyéb algebrai tulajdonságaira is magyarázatot ad. A valós számoknál is használjuk a maradékos osztást: pl. $b = 2\pi$ a trigonometrikus függvényeknél.) □

Nagyméretű számoknál már a maradékos osztás gyakorlati kiszámításának módján is érdemes elgondolkodnunk, erről részleteket pl. [MGy]-ben találunk.

A maradékos osztás legfőbb alkalmazása következő alfejezetben található:

## 4.2. Euklidesz algoritmusa

Meglepő módon az alábbi *eljárást* több mint 2300 éve ugyanígy írta le Euklidesz i.e. 300 körül. A görög matematika elsősorban geometria volt, így nem meglepő, hogy Euklidesz algoritmusát *tetszőleges szakaszokra* írta le – azaz Euklidesz *tetszőleges pozitív valós számokra definiálta és határozta meg algoritmikusan a legnagyobb közös osztót.*

A módszer a modern számítógépek korában is nélkülözhetetlen.
**Alkalmazásai** például: relatív prímek ellenőrzésénél és keresésénél, törtműveleteknél (törtek egyszerűsítésekor), lineáris Diophantikus egyenleteknél, lineáris kongruenciáknál és kongruenciarendszereknél (Kínai Maradéktétel), számok $(\mathrm{mod}\ m)$ multiplikatív inverzének kiszámításakor, . . . – ezeket a következő fejezetekben ismerhetjük meg.

**4.3. Algoritmus. Euklideszi algoritmus***lnko*$(a, b)$ **meghatározására:**

Adott $a, b \in \mathbb{Z}$ számokra ismételten alkalmazzuk a maradékos osztás tételét: ha $|a| \geq |b|$ akkor osszuk el az $a$ számot $b$-vel, majd $b$-t a maradékkal, stb. mindig az osztót a maradékkal.

Azaz legyen

$$\begin{aligned}
a &= b \cdot q_1 + r_1, & 0 < r_1 < |b|,\\
b &= r_1 \cdot q_2 + r_2, & 0 < r_2 < r_1,\\
r_1 &= r_2 \cdot q_3 + r_3, & 0 < r_3 < r_2,\\
r_2 &= r_3 \cdot q_4 + r_4, & 0 < r_4 < r_3,\\
&\vdots\\
r_{i-2} &= r_{i-1} \cdot q_i + r_i\ , & 0 < r_i < r_{i-1}\ ,\\
&\vdots\\
r_{m-2} &= r_{m-1} \cdot q_m + r_m\ , & 0 < r_m < r_{m-1}\ ,\\
r_{m-1} &= r_m \cdot q_{m+1}\,\bullet
\end{aligned}$$

Az algoritmus akkor áll meg, ha $0$ maradékot kapunk, azaz $r_{m+1} = 0$.
**Ekkor**

$$lnko(a, b) = r_m$$

vagyis „a legutolsó, nem nulla maradék”. □

**4.4. Megjegyzés.** Az Euklideszi algoritmus előnye *nem csak* a gyorsasága! Ha vele a legnagyobb közös osztót az argumentumok prímtényezős felbontása nélkül is ki lehet számolni, akkor prímtesztelő és -felbontó algoritmusoknál a vizsgálandó („prím vagy összetett?”) számok lnko-ját is ki tudjuk számolni! Erre pedig számtalanszor lesz szükségünk a 8. „*Prímtesztelés és számok felbontása*” fejezetben! □

**4.5. Jelölés.** Mivel a későbbi gyakorlati számításoknál szükségünk lesz a maradékok ($r_i$) és a hányadosok ($q_i$) megkülönböztetésére, ezért az $r_i$ *maradékokat* $\langle \dots \rangle$ zárójelekbe tettük: $\langle r_i \rangle$. □

**4.6. Példa. lnko(9867,8855) meghatározása:**

$$\begin{aligned}
< 9867 > &= < 8855 > *1 + < 1012 >\\
< 8855 > &= < 1012 > *8 + < 759 >\\
< 1012 > &= < 759 > *1 + < 253 >\\
< 759 > &= < 253 > *3 + < 0 >
\end{aligned}$$

tehát

$$lnko(9867, 8855) = 253. \quad \square$$

**4.7. Példa. lnko(5 170 549, 4 195 813) meghatározása:**

$$\begin{aligned}
&< 5170549 > \;=\; < 4195813 > *1+ \;\; < 974736 > \\
&< 4195813 > \;=\; < 974736 > *4+ \;\; < 296869 > \\
&< 974736 > \;=\; < 296869 > *3+ \;\; < 84129 > \\
&< 296869 > \;=\; < 84129 > *3+ \;\; < 44482 > \\
&< 84129 > \;=\; < 44482 > *1+ \;\; < 39647 > \\
&< 44482 > \;=\; < 39647 > *1+ \;\; < 4835 > \\
&< 39647 > \;=\; < 4835 > *8+ \;\; < 967 > \\
&< 4835 > \;=\; < 967 > *5+ \;\; < 0 >
\end{aligned}$$

tehát

$$lnko(5\ 170\ 549,\ 4\ 195\ 813) = 967. \quad \square$$

További kidolgozott példákat és alkalmazásokat találunk az [SzI1] feladatgyűjtemény 50–52., ill. 118–127. oldalain, továbbá a jegyzethez mellékelt EUKLDIO2D.EXE program segítségével gyakorolhatjuk az algoritmust.

Láthatjuk az előző példákban, hogy az algoritmus néhány egyszerű lépés után megáll, míg pl. a fenti hétjegyű számok törzstényezős felbontása sokkal tovább tartana.

Az algoritmusokról szóló 1.1. "*Alapfogalmak*" alfejezetben általánosságban említettük a „jó" algoritmusok legfontosabb jellemzőit, most ezen szempontok alapján részletesen megvizsgáljuk az Euklideszi algoritmust:

**1.)** megáll-e egyáltalában az algoritmus *minden* input esetében,
**2.)** mennyi idő múlva áll meg (másodperc törtrésze vagy évmilliók?),
**3.)** helyes eredményt ad-e *minden* inputnál,
**4.)** milyen nehéz/bonyolult, („mennyibe kerül"),
**5.)** milyen más problémákhoz lehet még alkalmazni.

Az Euklideszi algoritmus esetében mindegyik kérdésre meglepően jó választ kapunk.

**1.)** Az eljárás természetesen véges sok lépésben véget ér mert

$$|b| > r_1 > r_2 > \cdots > r_i > r_{i+1} > \cdots > 0$$

és *pozitív egész* számok csökkenő sorozata nem lehet végtelen. Ezt az „elvet" hívják **descente infinie** (végtelen leszállás, francia) **elv** -nek.

A fenti példában $4\ 195\ 813$-nál kevesebb maradékos osztást kellett végeznünk. Szerencsére ez mindig így van:

**3.)**

**4.8. Tétel** (Lamé)**.** *Az Euklideszi algoritmus legfeljebb annyi lépésig tart, mint amennyi $b$ számjegyei számának ötszöröse, azaz*

$$m \leq 5 \cdot \log_{10} |b|. \quad \square$$

*(Gabriel Lamé (1795-1870) francia matematikus. )*

**Bizonyítás.** A bizonyítás megtalálható például [CsL]-ben, most csak vázoljuk. Az alapötletet: az 4.3. Algoritmus talán akkor tart a legtovább, ha a benne szereplő $q_i$ együtthatók a legkisebbek. Azonban $q_i = 1$ esetén $(r_i)$ éppen a Fibonacci sorozat (visszafelé írva), tehát a Binet-formula szerint (ld.pl. [SzI2])

$$b = \frac{1}{\sqrt{5}} \cdot \left( \left( \frac{1+\sqrt{5}}{2} \right)^m - \left( \frac{1-\sqrt{5}}{2} \right)^m \right) \tag{4.3}$$

ahol $m$ az algoritmus lépésszáma, ahonnan

$$m \approx \mathcal{O}\left(\log\left(b\right)\right).$$

■

**4.9. Megjegyzés.** Lamé fenti tétele szerint **az Euklideszi algoritmus lineáris**, vagyis az input méretének növelésével a futásidő CSAK lineárisan (egyenes arányban) növekszik, minden ilyen algoritmus pedig a lehető **leggyorsabb:** a futásidő lényegében az input beolvasása. □

[CLR] 33.2-5. Gyakorlata szerint (701.old.) az Euklideszi algoritmus futásideje $a > b > 0$ esetén $1 + \log_\tau (b)$ ami csökkenthető

$$1 + \log_\tau \left( \frac{b}{lnko\,(a,b)} \right)$$

-ra, ahol $\tau = \frac{\sqrt{5}+1}{2}$.

**2.)**

**4.10. Tétel** (Euklideszi algoritmus helyessége)**.**

$$r_m = lnko(a,b).$$

**Bizonyítás.** A tétel könnyen bizonyítható teljes indukcióval *alulról felfelé:* $r_m = lnko(r_m, r_{m-1}) = lnko(r_{m-1}, r_{m-2}) = lnko(r_{m-2}, r_{m-3}) = \cdots = lnko(r_i, r_{i-1}) = \cdots = lnko(a,b)$. ■

A **4.)** kérdésre már válaszoltunk.

**5.)** A legfontosabb alkalmazásokat a fejezet elején már felsoroltuk.

**4.11. Állítás.** *$lkkt\,(a,b)$, $lnko(a_1,\ldots,a_t)$ és $lkkt\,(a_1,\ldots,a_t)$ is meghatározhatók az Euklideszi algoritmus többszöri alkalmazásával (ezeket már az előző fejezetekben megismertük).* □

**4.12. Példa.** Mennyi $lnko(39\,137\,563,\,15\,836\,693,\,37\,219\,177) = \;\;?$
**Megoldás:** $lnko(39\,137\,563,\,15\,836\,693) = 39\,493,$
$lnko(37\,219\,177,\,39\,493) = 541,$
így $lnko(39\,137\,563,\,15\,836\,693,\,37\,219\,177) = \mathbf{541}.$ □

Az [SzI1] feladatgyűjtemény 50. ill. 119-120. oldalain még sok, részletesen kidolgozott példát találunk.

(A XX. században „természetesen” felfedeztek gyorsabb és egyszerűbb algoritmusokat az $lnko$ kiszámítására, pl. [KD] 2.kötet 4.5.2. alfejezetében megtalálhatjuk *J. Stein* (1961) algoritmusát, amely csak kivonást és felezést (bináris vessző áthelyezése) használ.)

# 5. fejezet

# Lineáris Diophantoszi egyenletek

*Diophantosz (Kr.u.250) görög matematikus* foglalkozott egyenletek *egész* gyökeivel, ezért hívjuk általában az olyan egyenleteket *Diophantoszi*-nak, amelyeknek az egész gyökeit keressük.

Most csak a *lineáris* Diophantoszi egyenletekkel foglalkozunk.

**5.1. Definíció.** Legyenek adottak az $a_1, \ldots, a_n, c \in \mathbb{Z}$ számok, és keresendők olyan $x_1, \ldots, x_n \in \mathbb{Z}$ *egész számok, melyekre*

$$a_1 x_1 + \cdots + a_n x_n = c. \tag{5.1}$$

A fenti egyenletet ***n*-változós lineáris Diophantoszi-** (vagy **Diophantikus**) **egyenlet**nek hívjuk. □

Nagyon egyszerű az alábbi tétel „szükséges” fele:

**5.2. Tétel.** *Az (5.1) egyenlet* ***akkor és csak akkor*** *oldható meg, ha*

$$lnko\,(a_1, \ldots, a_n) \mid c\,. \tag{5.2}$$

**Bizonyítás.** A fenti (5.2) feltétel *szükséges*ségét nagyon egyszerűen beláthatjuk: $d = lnko\,(a_1, \ldots, a_n)$ esetén nyilván

$$d \mid a_1 x_1 + \cdots + a_n x_n$$

hiszen $x_1, \ldots, x_n \in \mathbb{Z}$ *egész számok.*

A (5.2) feltétel *elégséges*sége a következő fejezetekben ismertetett megoldási algoritmusokból (azok létezéséből) következik. ■

## 5.1. $ax + by = c$ egyenletek

Ebben az alfejezetben az

$$ax + by = c \tag{5.3}$$

kétismeretlenes Diophantikus egyenletet oldjuk meg, ahol $a, b, c \in \mathbb{Z}$ adott számok, teljesül az

$$lnko(a, b) \mid c$$

feltétel, és keresendő $x, y \in \mathbb{Z}$.

Elegendő először (5.3) helyett az

$$ax + by = lnko(a, b) \tag{5.4}$$

egyenletet megoldanunk, utána (5.3) már könnyen ($\frac{c}{d}$-vel beszorozva) megoldható, ahol $d = lnko(a, b)$.

**5.3. Tétel.** *Minden $a$ és $b$ egész számhoz léteznek olyan $u_0$ és $v_0$ egész számok, hogy*

$$lnko(a, b) = a \cdot u_0 + b \cdot v_0. \tag{5.5}$$

Az alábbi bizonyítás egyben **algoritmust** is ad $u_0$, $v_0$ (és így $x$, $y$) megtalálására, lényegében az Euklideszi algoritmus „kibővítésére" van szükségünk:

**5.4. Algoritmus. (Kiterjesztett Euklideszi Algoritmus)**
Az (5.5) **Diophantikus egyenlet megoldása**:
Használjuk fel a (4.3) algoritmus sorait, alulról a második sortól kezdve, **alulról felfelé** sorrendben (osztanunk nem kell). Ehhez részletesebben meg kell néznünk az Euklideszi algoritmus utolsó sorait:

$$\begin{aligned}
a &= b \cdot q_1 + r_1 \\
&\vdots \\
r_{m-4} &= r_{m-3} \cdot q_{m-2} + r_{m-2} \\
r_{m-3} &= r_{m-2} \cdot q_{m-1} + r_{m-1} \\
r_{m-2} &= r_{m-1} \cdot q_m + \mathbf{r}_m \\
r_{m-1} &= r_m \cdot q_{m+1} + 0.
\end{aligned}$$

Tehát:

$lnko(a, b) = r_m = r_{m-2} - r_{m-1} \cdot q_m = r_{m-2} - (r_{m-3} - r_{m-2} \cdot q_{m-1}) \cdot q_m =$
$= r_{m-3} \cdot u_{m-2} + r_{m-2} \cdot v_{m-2} = r_{m-3} \cdot u_{m-2} + (r_{m-4} - r_{m-3} \cdot q_{m-2}) \cdot v_{m-2} =$
$= r_{m-4} \cdot u_{m-3} + r_{m-3} \cdot v_{m-3} = \cdots$
$\ldots$
$= \mathbf{a} \cdot \mathbf{u}_0 + \mathbf{b} \cdot \mathbf{v}_0.$ □

**5.5. Megjegyzés.** Az $u_i$, $v_i$ számokat nem kell képlettel kiszámítanunk, hiszen menet közben „automatikusan" megkapjuk őket, mint a következő példában láthatjuk.
(Amint a 4.5. pontban megjegyeztük: a számolásban meg kell különböztetnünk az $r_i$ *maradékokat, ezeket* $\langle \ldots \rangle$ zárójelekkel emeljük ki: $\langle r_i \rangle$.) □

**5.6. Példa.** Oldjuk meg a

$$9867x + 8855y = 759 \tag{5.6}$$

egyenletet.

**Megoldás:** van megoldás, mert $lnko(9867, 8855) = 253 \mid 759$.

A 4.6. Példa számolásai alapján (a maradékokat $<\ldots>$ zárójelekbe tettük, mint ezt a 4.5. Jelölésben definiáltuk):

lnko(9867,8855) = **253** = 1·<1012> + (-1)·<759> =
= 1·<1012> + (-1)·(<8855>-8·<1012>) =
= (-1)· <8855>+9·<1012> = (-1)·<8855>+9·(<9867>-1·<8855>) =
= 9·<9867> -10·<8855> ,

ahonnan

$$u_0 = 9, \quad v_0 = -10.$$

Mivel $759/253 = 3$, ezért az (5.6) egyenlet egyik megoldása

$$u_0 = 9 \cdot 3 = \mathbf{27}, \quad v_0 = -10 \cdot 3 = \mathbf{-30}.$$

Az egyenletnek van még több gyöke, amiket az 5.10. Tételben adunk meg. □

A mellékelt EUKLDIO2D.EXE program segítségével gyakorolhatjuk a fenti módszert.

A [SzI1] Feladatgyűjteményben (52–54., 127–134. old) sok, részletesen kidolgozott feladatot találunk a fenti algoritmusra és a lineáris Diophantikus egyenletek alkalmazásaira.

**5.7. Megjegyzés.** Az 5.4. Algoritmusból kiolvasható a

$$\left\{ \begin{array}{l} u_{m-1} = \quad 1 \\ v_{m-1} = -q_m \end{array} \right. , \quad \left\{ \begin{array}{l} u_i = -v_{i+1} \\ v_i = -u_{i+1} - q_{i+1} \cdot v_{i+1} \end{array} \right. \quad (i = m-2, \ldots, 0) \tag{5.7}$$

vagy másképpen

$$\left\{ \begin{array}{l} u_{i+1} = q_{i+1} \cdot u_i + u_{i-1} \\ v_{i+1} = -u_i \end{array} \right.$$

rekurzív összefüggés, de mint láttuk, nincs szükségünk rá.

Tehát ebben az alfejezetben sikerült bebizonyítanunk az alábbi tételt:

**5.8. Tétel.** *Az*

$$ax + by = c \tag{5.8}$$

*Diophantikus egyenletnek* ***pontosan akkor*** *létezik megoldása, ha*

$$lnko(a, b) \mid c\,. \quad \square$$

Fontos kérdés az algoritmus sebessége, erre megnyugtató választ tudunk adni:

**5.9. Megjegyzés.** Lamé 4.8. Tétele ismét hasznos számunkra: az (5.1) egyenleteket megoldó fenti algoritmus is **lineáris**, vagyis (problémánkra) a lehető leggyorsabb. □

Végül keressük meg az (5.3) egyenletek *összes* (”*általános*”) megoldását!
Ha $u_0$ és $v_0$ egy megoldása az

$$a \cdot u_0 + b \cdot v_0 = lnko(a, b)$$

egyenletnek, akkor könnyen látható, hogy

$$\mathbf{x}_0 := u_0 \cdot \frac{c}{lnko(a, b)}, \quad \mathbf{y}_0 := v_0 \cdot \frac{c}{lnko(a, b)}$$

egy megoldása az (5.3) egyenletnek.

Az *összes* megoldást pedig az alábbi tétel segítségével kapjuk meg:

**5.10. Tétel.** *Ha az (5.8) egyenletnek létezik egy $(x_0, y_0)$ megoldása, akkor az **összes gyök**:*

$$\mathbf{x} = x_0 + k \cdot \frac{lkkt(a, b)}{a}, \qquad \mathbf{y} = y_0 - k \cdot \frac{lkkt(a, b)}{b} \qquad (k \in \mathbb{Z}) \tag{5.9}$$

*vagy másképpen*

$$\mathbf{x} = x_0 + k \cdot \frac{b}{lnko(a, b)}, \qquad \mathbf{y} = y_0 - k \cdot \frac{a}{lnko(a, b)} \qquad (k \in \mathbb{Z}) \tag{5.10}$$

*($k \in \mathbb{Z}$ tetszőleges szám).*

**Bizonyítás.** A tétel könnyen bizonyítható: vizsgáljuk meg az (5.8) egyenlet két gyöke közötti különbséget (házi feladat). ■

Térjünk vissza kicsit az 5.4. Algoritmushoz. Gyors, érthető, áttekinthető. A baj csak az, hogy sok memóriát igényel, hiszen az összes $q_i$ és $r_i$ maradékot el kell tárolnunk, mivel az algoritmus a második menetben az egyenletek sorait visszafelé haladva is felhasználja. Az egyenleteket közelebbről megvizsgálva ez a kettősség kiküszöbölhető és az algoritmus is kicsit gyorsítható. Az alábbi módosítás implementálása és bizonyítása megtalálható például [KD] 1.kötet 37. oldalán.

**5.11. Algoritmus.** Az $ax + by = lnko\,(a, b)$ (5.5) egyenlet megoldására.
(Az 5.4. Algoritmus jelöléseit használjuk.)
Legyenek $\xi_{-1} := 1, \zeta_{-1} := 0; \quad \xi_0 := 0, \zeta_0 := 1,$
majd az $r_{i-2} = r_{i-1} \cdot q_i + r_i$ vagyis $r_i := r_{i-2} - r_{i-1} \cdot q_i$ sor mellett a következőt is számítsuk ki:

$$\xi_i := \xi_{i-2} - \xi_{i-1} \cdot q_i, \quad \zeta_i := \zeta_{i-2} - \zeta_{i-1} \cdot q_i\,.$$

Ekkor $\xi_m, \zeta_m$ megadja az $ax + by = lnko\,(a, b)$ egyenlet egy megoldását. □

**5.12. Példa.** Oldjuk meg az

$$9867x + 8855y = 759 \tag{5.11}$$

egyenletet (ld. 4.6. és 5.6. Példák).

$\xi_{-1} := 1, \quad \zeta_{-1} := 0;$
$\xi_0 := 0, \quad \zeta_0 := 1,$

$$\begin{array}{ll}
< 9867 > = < 8855 > * \mathbf{1} + < 1012 > & \xi_1 = 1 - 0 * \mathbf{1} = 1 \\
 & \zeta_1 = 0 - 1 * \mathbf{1} = -1 \\
< 8855 > = < 1012 > * \mathbf{8} + < 759 > & \xi_2 = 0 - 1 * \mathbf{8} = -8 \\
 & \zeta_2 = 1 - (-1) * \mathbf{8} = 9 \\
< 1012 > = < 759 > * \mathbf{1} + < 253 > & \xi_3 = 1 - (-8) * \mathbf{1} = 9 \\
 & \zeta_3 = -1 - 9 * \mathbf{1} = -10 \\
< 759 > = < 253 > * 3 + < 0 > & VÉGE
\end{array}$$

tehát $lnko\,(9867x + 8855) = 253$, $\xi = \xi_3 = 9$ és $\zeta = \zeta_3 = -10$ egy megoldása az $9867\xi + 8855\zeta = 253 = lnko$ egyenletnek,
vagyis

$$x_0 := \xi_3 \cdot \frac{759}{253} = 27 \quad \text{és} \quad y_0 := \zeta_3 \cdot \frac{759}{253} = -30$$

egy megoldása az (5.11) egyenletnek. □

Az 5. "*Kongruenciák és maradékosztályok*" és 6. "*Kínai maradéktétel...*" fejezetekben a lineáris Diophantikus egyenleteket több, elméleti és számítástechnikai probléma megoldására tudjuk felhasználni.

[LG] 100-104.oldalain az Euklideszi algoritmus érdekes alkalmazását találjuk *polinomok* gyökei számának meghatározására egy adott $[a, b]$ intervallumon.

## 5.2. $a_1x_1 + \cdots + a_nx_n = c$ egyenletek

Az (5.1) egyenlet összes megoldását az Euklideszi algoritmus többszöri alkalmazásával ($n$-re történő teljes indukcióval) megkaphatjuk, ami a (5.2) tétel „elégséges" felét is igazolja. Az [SzI1] feladatgyűjteményben erre is találunk kidolgozott példákat (53-54. ill. 132-134.oldalak).

Most csak az $n = 3$ esetet mutatjuk meg röviden.

Megoldandó tehát az

$$ax + by + cz = m \tag{5.12}$$

*háromismeretlenes* lineáris Diophantikus egyenlet, ahol $a, b, c, m, x, y, z \in \mathbb{Z}$ egész számok.

**(0)** Előrebocsátjuk, hogy a megoldhatóság *szükséges és elégséges* feltétele:

$$lnko(a, b, c) \mid m\,.$$

**(1)** Az Euklideszi algoritmussal megkeressük $d := lnko(a, b)$-t.

**(2)** az $ax + by = td \quad (t \in \mathbb{Z})$ egyenletek általános megoldása

$$x = t \cdot x_o + \frac{lkkt(a,b)}{a} \cdot k, \quad y = t \cdot y_o + \frac{lkkt(a,b)}{b} \cdot k \qquad (k \in \mathbb{Z}).$$

**(3)** Számítsuk ki $\delta := lnko(d,c)$ értékét (az Euklideszi algoritmussal).

**(4)** Oldjuk meg az $dt + cz = m$ egyenletet, melynek általános megoldása

$$t = \frac{m}{\delta} \cdot t_o + \frac{lkkt(d,c)}{d} \cdot \ell, \quad z = \frac{m}{\delta} \cdot z_o - \frac{lkkt(d,c)}{c} \cdot \ell \qquad (\ell \in \mathbb{Z}).$$

**5.13. Megjegyzés.** Tehát az

$$ax + by + cz = m$$

lineáris Diophantikus egyenletek általános megoldása:

$$\left\{\begin{array}{l} x = t \cdot x_o + \dfrac{lkkt(a,b)}{a} \cdot k \\ y = t \cdot y_o + \dfrac{lkkt(a,b)}{b} \cdot k \\ z = \frac{m}{\delta} \cdot z_o - \dfrac{lkkt(d,c)}{c} \cdot \ell \end{array}\right. \quad \textit{ahol} \quad \begin{array}{l} t = \dfrac{m}{\delta} \cdot t_o + \dfrac{lkkt(d,c)}{d} \cdot \ell \\ k, \ell \in \mathbb{Z}\,. \end{array}$$

**ahol** $\delta = lnko(a,b,c)$, továbbá a megoldhatóság *szükséges és elégséges* feltétele:

$$\delta \mid m \quad .$$

□

**5.14. Példa.** Oldjuk meg az $12x + 30y + 15z = 18$ egyenletet az egész számok körében.

**(0)** mivel $lnko(12,30,15) = 3 \mid 18$, ezért van gyöke az egyenletnek,

**(1)** $d = lnko(12,30) = 6$,

**(2)** a $12x + 30y = t \cdot 6$ egyenletek általános megoldása:

$$x = t \cdot (-2) + 5k, \quad y = t \cdot 1 - 2k,$$

**(3)** $\delta = lnko(d,c) = lnko(6,15) = 3$,

**(4)** a $6 \cdot t + 15 \cdot z = 18$ egyenlet általános megoldása:

$$t = 6 \cdot (-2) + 5\ell, \quad z = 6 \cdot 1 - 2\ell \quad (\ell \in \mathbb{Z}),$$

így az általános megoldás:

$$\left\{\begin{array}{l} x = t \cdot (-2) + 5 \cdot k \\ y = t \cdot 1 - 2 \cdot k \\ z = 6 \cdot 1 - 2\ell \end{array}\right. \quad \textit{ahol} \quad \begin{array}{l} t = -12 + 5 \cdot \ell \\ k, \ell \in \mathbb{Z} \end{array}$$

azaz

$$\begin{cases} x = (-12 + 5\ell) \cdot (-2) + 5k & = -10\ell + 5k + 24 \\ y = (-12 + 5\ell) - 2k & = \quad 5\ell - 2k - 12 \\ z = 6 - 2\ell & = -2\ell + 6 \end{cases} \qquad \textit{ahol} \quad k, \ell \in \mathbb{Z}.$$

□

# 6. fejezet

# Kongruenciák és maradékosztályok

A periodikusan ismétlődő jelenségeknél általában nem az ismétlődések száma, hanem a legvégén kapott *maradék* a fontos (hét napjai, trigonometriában $2k\pi$ utáni maradék azaz „hol jövök ki a körforgalomból?", csomagolás utáni maradék, a szám legutolsó jegye vagy néhány legutolsó jegye, túlcsordulás egész számoknál a számítógépben, „mindenkinek van-e párja?" a tánciskolában, stb.).

A most következő fejezetben *egész számok* egész számmal való osztási maradékait fogjuk részletesen megvizsgálni.

A [SzI1] Feladatgyűjtemény 4.2. alfejezetében rengeteg kidolgozott példát találunk oszthatósági szabályokról, maradékokról, kongruenciákról és alkalmazásaikról.

## 6.1. Kongruenciák

(*kongruencia* (lat.) = *megegyezés, megfelelés, egybevágóság.)*

Az osztási maradékok vizsgálatánál hasznos az alábbi tömör jelölés:

**6.1. Definíció.** Tetszőleges $a, b, m \in \mathbb{Z}$, $m \neq 0$ egész számokra jelölje

$$a \equiv_m b \quad \text{vagy} \quad a \equiv b \pmod{m}$$

(olvasd: "$a$ **kongruens** $b$**-vel modulo** $m$") az „$a$ és $b$ **ugyanazt a maradékot adják** $m$-el elosztva" relációt (összefüggést). □

Bár a *fenti* definíció fejezi ki a kongruencia lényegét, matematikai környezetben (bizonyításokhoz) az *alábbi* definíció a hasznos:

**6.2. Definíció.** Legyen $m \in \mathbb{Z}$ tetszőleges (rögzített) egész szám, $m \neq 0$. Ekkor tetszőleges $a, b \in \mathbb{Z}$ számokra legyen

$$a \equiv_m b \quad \text{vagy} \quad a \equiv b \pmod{m} \tag{6.1}$$

pontosan akkor ha

$$m \mid a - b. \tag{6.2}$$

$m$-et a kongruencia **modulusá**nak nevezzük. □

Szerencsére a fenti két definíció ekvivalens (*azonos értékű*, lat.): a (6.2) összefüggés pontosan azt jelenti hogy "$a$ és $b$ ugyanazt a maradékot adják $m$-mel elosztva" (HF).

**6.3. Megjegyzés.** **(i)** Az $m = \pm 1$ eseteket érdemes külön is megvizsgálnunk: bármely $a, b \in Z$ számokra $a \equiv b \pmod{\pm 1}$. Általában $m \geq 2$ és szigorúan $m \neq 0$!
**(ii)** $\equiv_m$ pontosabb elnevezése számelméleti kongruencia, hiszen a geometriai egybevágóság sok élő nyelven (latin eredete miatt) szintén kongruencia, és van általános (absztrakt) algebrai kongruencia is (ez utóbbi: művelettartó ekvivalencia-reláció).
**(iii)** $=$ és $\equiv$ nem keverhetők össze: tetszőleges $a, b, m \in Z$, $m \neq 0$ egész számokra $a = b$-ből következik $a \equiv_m b$ de megfordítva általában nem!
Hát persze: az $=$ megkülönböztet minden $a, b$ egész számot, míg $\equiv_m$ nagyon sok (végtelen sok) számot "összemos". Továbbmenve: nyilvánvaló az alábbi következtetés:

**6.4. Állítás.** *$m \mid n$ esetén $a \equiv_n b$-ből következik $a \equiv_m b$, de megfordítva (általában) nem.* □

**6.5. Megjegyzés.** **(folytatás)** vagyis $\pmod{n}$ sokkal több számot különböztet meg mint $\pmod{m}$ ha $m \mid n$. Algebrai nyelven ezt úgy mondják, hogy $\pmod{n}$ finomabb osztályozás/reláció mint $\pmod{m}$, ami pedig durvább. Nyilvánvalóan $\pmod{\pm 1}$ a legdurvább és $=$ azaz $\pmod{\infty}$ a legfinomabb. □

Az $a \equiv_m b$ jelölést legtöbbször akkor használjuk, ha $a \in \mathbb{Z}$ tetszőleges és $0 \leq b < m$, hiszen $a$ maradékára $\pmod{m}$ vagyunk kíváncsiak és ezt jelöljük $b$-vel.

Így tulajdonképpen egy *művelet*et definiáltunk: az $a \in \mathbb{Z}$ szám $m$-el való osztási maradékát. Nagy $m$ modulusoknál $b$ is nagy lehet, ezért érdemes *negatív* maradékokra is gondolni (mert ekkor $m - b$ kicsi).

**6.6. Definíció.** Legyen $m \in \mathbb{Z}$ tetszőleges (rögzített) egész szám, $m \neq 0$. Ekkor tetszőleges $a \in \mathbb{Z}$ számra
**(i) jelölje**

$$a \pmod{m} \quad \text{vagy} \quad (a \bmod m)$$

azt az (egyetlen) $b < m$ *nemnegatív* számot, amelyre $b \equiv_m a$ teljesül,
**(ii) jelölje**

$$a \ (MOD\, m) \quad \text{vagy} \quad (a\ MOD\, m)$$

azt a *legkisebb abszolút értékű* $b$ számot (tehát $-\frac{m}{2} < b \leq \frac{m}{2}$), amelyre $b \equiv_m a$ teljesül. □

Vigyázat: a fenti definíció érzékeny a kis- és nagybetűkre: $\pmod{m}$ és $(MOD\, m)$ nem ugyanaz!

A legtöbb számítógépes programozási nyelvben is van beépített maradékképző művelet (/eljárás/függvény), a zsebszámlógépeken sajnos nincs ilyen gomb.

**6.7. Megjegyzés.** A fenti jelölésekkel $x \equiv_m y$ (6.1) helyett

$$(x \bmod m) = (y \bmod m)$$

is írható: a két írásmód ekvivalens. □

Házi feladat a következő összefüggések ellenőrzése:

**6.8. Állítás.** *Rögzített $m \in \mathbb{Z}$, $m \neq 0$ számra* $\equiv_m \subset \mathbb{Z} \times \mathbb{Z}$ *ekvivalencia-reláció (vagyis: reflexív, szimmetrikus és tranzitív bináris reláció).* □

**6.9. Tétel.** *Tetszőleges (*rögzített*) $m \in \mathbb{Z}$, $m \neq 0$ számra, valamint bármely $a, b, c, d \in \mathbb{Z}$ számokra*
***ha*** $a \equiv_m b$
*és* $c \equiv_m d$
***akkor*** $a \pm c \equiv_m b \pm d$
*és* $a \cdot c \equiv_m b \cdot d$ □

A fenti összefüggések miatt hívhatjuk $\equiv_m$-t kongruenciának (művelettartó ekvivalencia reláció).

**6.10. Megjegyzés.** **(i)** Az iskolában tanult "*oszthatósági*" szabályok is a fenti 6.9. Tétel következményei.

Például a 11-gyel oszthatóság szabálya azon alapszik, hogy

$$10^j \equiv (-1)^j \pmod{11}$$

tehát egy $a_k a_{k-1} \dots a_1 a_0$ *számjegyekkel, tízes számrendszerben leírt $n$ szám*

$$n = \overline{a_k a_{k-1} \dots a_1 a_0}^{(10)} := \sum_{j=0}^{k} 10^j \cdot a_j$$

11-gyel való osztási maradéka

$$n \equiv \sum_{j=0}^{k} (-1)^j \cdot a_j \pmod{11}$$

vagyis kapjuk a jólismert *szabályt:*

**„Egy tízes számrendszerben felírt szám pontosan akkor osztható 11-gyel, ha a számjegyeit váltakozó előjellel összeadva a kapott összeg osztható 11-gyel."** (A váltakozó előjelek a 0 számjegyekre is vonatkoznak, például $n = 1032002$ maradéka $\equiv 1 - 0 + 3 - 2 + 0 - 0 + 2 \equiv 4 \pmod{11}$.)

**(ii)** Hasonló kérdés: „milyen számjegyekre végződik a megadott HATALMAS kifejezés?", hiszen ha $n \in \mathbb{N}$ legutolsó $\ell$ számjegyét kérdezzük, akkor valójában a $\left(\mathrm{mod}\ 10^\ell\right)$ maradékra vagyunk kíváncsiak. A fenti 6.9. Tétel alapján megint egy jólismert *szabályt* kapunk:

**„A végeredmény utolsó $\ell$ jegyének meghatározásához mindössze csak a tagok/tényezők utolsó $\ell$ jegyeit kell figyelembe vennünk."**

A [SzI1] Feladatgyűjteményben sok, részletesen kidolgozott feladatot találunk a fenti *szabályok* gyakorlására és alkalmazására. □

A fenti 6.9. Tétel hasznát a következőképpen foglalhatjuk össze:

**6.11. Megjegyzés.** Ha egy nagyméretű kifejezés kiértékelésénél (nagy számolásnál) *csak* a végeredmény $(\mathrm{mod}\ m)$ *maradéka* érdekel minket, *rögzített* $m$ modulus esetén, akkor *minden lépésben* vehetjük/vegyük a részeredmény maradékát és csak a (kisméretű) maradékokkal kell tovább számolnunk. Vagyis egyetlen lépésben sem kell nagyméretű számokkal bajlódnunk. (Ezt hívják **moduláris aritmetiká**nak.)
Ezt nem csak általános- és középiskolai feladatoknál, hanem a jelen és a későbbi fejezetekben is használhatjuk, ennek látványos alkalmazása például a 6.6. „Nagy kitevőjű hatványozás" alfejezet. □

**6.12. Példa.** Mennyi maradékot ad a $132465+46587\cdot 83152\cdot 731052-208645^5$ kifejezés 753-mal osztva?
**Megoldás:** mindegyik tényezőnek külön-külön vesszük a 753-mal való osztási maradékát, és a számolás minden lépésében is a részeredmények helyett 753-mal való osztási maradékukat tekintjük:

$$\begin{aligned}
&132465 + 46587 \cdot 83152 \cdot 731052 - 208645^5 \equiv \\
&\quad \equiv 690 + 654 \cdot 322 \cdot 642 - 64^5 \equiv \\
&\quad \equiv 690 + 654 \cdot 206724 - 64^2 \cdot 64^3 \equiv \\
&\quad \equiv 690 + 654 \cdot 402 - 331 \cdot 100 = \\
&\quad \equiv 230498 \equiv 80 \ (\mathrm{mod}\ 753). \quad \square
\end{aligned}$$

Sok gyakorló feladatot részletes megoldásokkal találhatunk [SzI1] 39-45. ill. 97-109. oldalain. □

**6.13. Megjegyzés. VIGYÁZAT:** *páratlan* modulus esetén általában már **nem igazak** az alábbi, jól megszokott állítások:
"*páros*±*páros*=*páros*", "*páratlan*±*páratlan*=*páros*", **...**, $(\mathrm{mod}\ m)$
"*páros*·*páros*=*páros*", "*páratlan*·*páratlan*=*páratlan*" $(\mathrm{mod}\ m)$,
például $6+4\equiv 1$, $6\cdot 2\equiv 3$ de $6\cdot 4\equiv 6\ (\mathrm{mod}\ 9)$, stb.
Ezt a kérdést a 6.3. "*Elsőfokú kongruencia-egyenletek*" alfejezet eredményei alapján tudjuk teljességgel megvizsgálni. □

Végezetül *különböző* modulusokra vonatkozó összefüggéseket említünk meg:

**6.14. Tétel.** ***(i)*** Tetszőleges $m_1, m_2 \in \mathbb{Z}$ $(m_1 m_2 \neq 0)$ *modulusokra:*
*ha* $x \equiv y\ (\mathrm{mod}\ m_1)$ *és* $x \equiv y\ (\mathrm{mod}\ m_2)$ *akkor* $x \equiv y\ (\mathrm{mod}\ lkkt(m_1, m_2))$.
***(ii)*** *ha* $ac \equiv bc\ (\mathrm{mod}\ m)$ *és* $d = lnko(c, m)$ *akkor* $a \equiv b\ \left(\mathrm{mod}\ \frac{m}{d}\right)$. □

## 6.2. Maradékosztályok

Az alábbiakban a maradékok közötti műveleteket vizsgáljuk, algebrai szempontok szerint. Nem kívánunk precíz absztrakt fogalmakat és tételeket használni, mindössze a téma rövid bemutatása a célunk.

Az előző fejezet 6.9. Tétele és az utána következő Megjegyzések szerint elegendő a számok maradékaival végeznünk a műveleteket – ezt összegezzük a következő definíciókban és tételekben.

Jelen fejezet végéig rögzítsünk egy *tetszőleges* $n \geq 1$ *pozitív egész szám*ot.

**6.15. Jelölés.** Jelölje $\mathbb{Z}_n$ a $(\mathrm{mod}\ n)$-*összes* maradékok halmazát:

$$\mathbb{Z}_n := \{0, 1, \ldots, n-1\}\,.$$

Különböző algebrai meggondolások miatt $\mathbb{Z}_n$ helyett használatos még a $(\mathbb{Z}/_{n\cdot\mathbb{Z}})$ jelölés is. □

**6.16. Megjegyzés.** Algebrai szempontból $\mathbb{Z}_n$ a $\equiv_n$ ekvivalenciareláció osztályait tartalmazza, erre nekünk most nem lesz szükségünk, de technikai okok miatt az alábbi Definícióra és szemléletre igen:

**6.17. Definíció.** Az $n$ db tetszőleges $\{a_1, \ldots, a_n\} \subseteq \mathbb{Z}$ számot **teljes maradékrendszer**nek hívjuk $(\mathrm{mod}\ n)$, ha az $a_i$ számok $(\mathrm{mod}\ n)$ maradékai az összes $\mathbb{Z}_n$-beli maradékot kiadják, mindegyiket pontosan egyszer. □

A későbbiekben többször fogjuk használni a következő eredményt:

**6.18. Tétel.** *Tetszőleges $n \in \mathbb{Z}$ számra ha $a_1, a_2, \ldots, a_n$* teljes *maradékrendszer $(\mathrm{mod}\ n)$, $c \in \mathbb{Z}$ egy tetszőleges és $b \in \mathbb{Z}$ egy $n$-hez* relatív prím *szám, akkor a*

$$ba_1 + c,\ ba_2 + c,\ \ldots,\ ba_n + c$$

*számok is teljes maradékrendszert alkotnak* $(\mathrm{mod}\ n)$. □

**6.19. Definíció. Összeadás és szorzás** a $\mathbb{Z}_n$ halmazon:
Tetszőleges $a, b \in \mathbb{Z}_n$ esetén legyen $a \oplus b$ és $a \odot b$ olyan elemei $\mathbb{Z}_n$-nek, amelyekre

$$a \oplus b :\equiv a + b \pmod{n}$$

és

$$a \odot b :\equiv a \cdot b \pmod{n}.$$

Ha nem okoz félreértést, egyszerűen csak a $+$ és $\cdot$ jeleket használjuk $\oplus$ és $\odot$ helyett. □

**6.20. Megjegyzés. (o)** A fenti két művelet jól meghatározott (definiált) az előző alfejezet 6.9. Tétele alapján:
ha

$$a \equiv a' \text{ és } b \equiv b' \pmod{n}$$

akkor

$$a \pm b \equiv a' \pm b' \quad \text{és} \quad a \cdot b \equiv a' \cdot b' \pmod{n}\,.$$

**(i)** Szintén a fenti tétel miatt a $\oplus$ és $\odot$ műveletek rendelkeznek a valós számok szokásos tulajdonságaival (asszociativitás, kommutativitás, disztributivitás), ezeket most nem írjuk fel. A továbbiakban csak a $+$ és $\cdot$ jeleket használjuk $\oplus$ és $\odot$ helyett.

(ii) A fentiek alapján $(\mathbb{Z}_n, +)$, $(\mathbb{Z}_n, \cdot)$ és $(\mathbb{Z}_n, +, \cdot)$műveletekre zárt halmazok, algebrai struktúrák, sőt $(\mathbb{Z}_n, \cdot)$ kommutatív félcsoport, $(\mathbb{Z}_n, +)$ Abel csoport és $(\mathbb{Z}_n, +, \cdot)$ kommutatív egységelemes gyűrű, azaz integritási tartomány. Ezeket a fogalmakat részletesebben a Függelékben vizsgáljuk.

(iii) A jelen „Algoritmikus számelmélet" könyvünk egésze a $(\mathbb{Z}_n, +, \cdot)$ gyűrűről szól. A valós együtthatós polinomok $R[x]$ halmaza (polinomgyűrű) is hasoló tulajdonságokkal bír, könyvünk nagy része $R[x]$-re is alkalmazható. □

Fontos gyakorlati probléma a műveletek „megfordíthatósága”: ellentett és inverzelem keresése.

Az összeadással nincs gond: tetszőleges $a \in \mathbb{Z}_n$ elemre $a + (-a) \equiv 0 \pmod{n}$, ahol nyilvánvalóan $(-a) = n - a$.

A szorzásnál sokkal bonyolultabb a probléma. Miért is baj, hogy például

$$2 \cdot 3 \equiv 0 \pmod{6} \quad ?$$

Ebből miért következik, hogy a

$$2 \cdot x \equiv 1 \pmod{6}$$

egyenletnek nincs megoldása?

**6.21. Definíció.** Legyen $(\mathcal{R}, +, \cdot)$ egy tetszőleges gyűrű (mondjuk $(\mathbb{Z}_n, +, \cdot)$). Egy tetszőleges $a \in \mathcal{R}$ (azaz $a \in \mathbb{Z}_n$) elem **multiplikatív inverze** egy olyan $a^{-1}$-nel jelölt eleme $\mathcal{R}$-nek ($\mathbb{Z}_n$-nek) amelyre

$$a \cdot a^{-1} = 1$$

vagyis $a^{-1}$ az

$$a \cdot x \equiv 1 \pmod{n} \tag{6.3}$$

kongruencia-egyenlet megoldása. □

Néha elhagyjuk a „multiplikatív” jelzőt ha nem okoz félreértést, de mivel létezik „additív” inverz is, ezért elővigyázatosnak kell lenünk.

Az (6.3) és hasonló egyenletek megoldásával (számelméleti módszerekkel) a következő fejezetben (ld. 6.36. Tétel), de előtte a következő, nagyon egyszerű, általános algebrai vizsgálatok is hasznosak számunkra.

**6.22. Állítás.** *Tetszőleges $\mathcal{R}$ gyűrűben minden $a$ elemnek legfeljebb egyetlen inverze lehet, vagyis az $a^{-1}$ inverzelem* egyértelmű.

**Vigyázat:** Most a $(\mathbb{Z}_n, \oplus, \odot)$ struktúrában vagyunk, és $a^{-1} \in \mathbb{Z}_n$, vagyis nem az iskolai szorzás $\cdot$ műveletéről van szó!

**Bizonyítás.** Ha $x$ és $y$ is inverze $a$-nak, akkor számítsuk ki kétféleképpen az $xay$ szorzatot:
egyrészt

$$xay = (xa)\, y = 1 \cdot y = y$$

másrészt

$$xay = x\,(ay) = x \cdot 1 = x$$

tehát $x = y$, Q.E.D.[1)] ■

[1)] Q.E.D. = quod erat demonstrandum = amit meg kellett mutatnunk (lat.)

**6.23. Definíció.** Legyen $(\mathcal{R}, +, \cdot)$ egy tetszőleges gyűrű (mondjuk $(\mathbb{Z}_n, +, \cdot)$), és jelölje $0$ a gyűrű összeadásra vonatkozó nullelemét.
**(i)** Az $a \neq 0$ elemet **nullosztó**nak vagy **zérusosztó**nak hívjuk, ha létezik olyan $b \neq 0$ elem, hogy $a \cdot b = 0$ . (Ekkor nyilván $b$ is nullosztó.)
**(ii)** Az $(\mathcal{R}, +, \cdot)$ gyűrűt **nullosztómentes**nek hívjuk ha nincsenek nullosztói, egyébként $(\mathcal{R}, +, \cdot)$ **nullosztós**. □

**6.24. Állítás.** *Ha $n$ összetett szám akkor $\mathbb{Z}_n$-ben léteznek zérusosztók. Részletesebben: $a \in \mathbb{Z}_n$ pontosan akkor zérusosztó ha* nem relatív prím *$n$-hez.*
***Bizonyítás:*** *házi feladat.* □

**6.25. Tétel.** *Ha $a \in \mathcal{R}$ nullosztó akkor nincs inverze.*

**Bizonyítás.** Ha $a$ nullosztó a $b$ párjával, $a^{-1}$ pedig inverrze $a$-nak, akkor számítsuk ki kétféleképpen a $baa^{-1}$ szorzatot:
egyrészt

$$baa^{-1} = (ba)\, a^{-1} = 0 \cdot a^{-1} = 0$$

másrészt

$$baa^{-1} = b\left(aa^{-1}\right) = b \cdot 1 = b$$

ami ellentmondás mert $b \neq 0$ tehát $a$ valóban nem lehet nullosztó és invertálható egyszerre.
■

**6.26. Definíció.** Az $(\mathcal{R}, +, \cdot)$ gyűrűt (például $(\mathbb{Z}_n, \oplus, \odot)$-et) **test**nek nevezzük, ha *minden* $a \neq 0$ elemének van multiplikatív inverze. □

**6.27. Következmény.** *Ha $n$ összetett szám akkor $\mathbb{Z}_n$-ben vannak elemek, amiknek nincs multiplikatív inverze: az $n$-hez* nem relatív prím *számoknak biztosan nincs.*
*Tehát $n$ összetett számra $\mathbb{Z}_n$ nem test.* □

**Vigyázat:** a fenti következtetések általában nem fordíthatók meg: nem minden *nemnullosztónak* van inverze (pl. $(\mathbb{Z}, +, \cdot)$-ben: az egész számok szokásos $+, \cdot$ műveleteire).

Az $n \in \mathbb{P}$ prímmodulusok különleges helyzetben vannak: nem csak a következő fejezet 6.36. Tétele, hanem az alábbi általános, nem túl nehéz algebrai eredmény miatt is:

**6.28. Tétel.** *Véges, nullosztómentes gyűrűben minden elemnek van multiplikatív inverze, vagyis az ilyen gyűrű test.*
Általánosabban: *ha egy $(\mathcal{S}, \cdot)$ félcsoport (például $(\mathbb{Z}_n, \cdot)$) minden $a, x, y$, $a \neq 0$ elemére: $ax = ay$-ból $x = y$ következik (lehet egyszerűsíteni $\mathcal{S}$-ben), akkor $\mathcal{S}$-ben van egységelem és minden elemnek van multiplikatív inverze (vagyis $\mathcal{S}$ csoport).* □

**6.29. Következmény.** *Bármely $p \in \mathbb{P}$ prímszámra $(\mathbb{Z}_p, +, \cdot)$ test, azaz minden $a \not\equiv 0 \pmod{p}$ elemre az $ax \equiv 1 \pmod{p}$ kongruenciának van megoldása.* □

Összetett modulusra kénytelenek vagyunk a modulushoz relatív prím számokkal foglalkozni.

**6.30. Definíció.** *Tetszőleges* $n \in \mathbb{Z}$ $(n \neq 0)$ számra $\mathbb{Z}_n$-nek $n$-hez relatív prím elemeinek *halmazát* **redukált maradékrendszer**nek nevezzük $(\mathrm{mod}\ n)$ és $\mathbb{Z}_n^*$-gal jelöljük.
Másképpen fogalmazva: $B = \{b_1, \ldots, b_k\} \subseteq \mathbb{Z}_m$ akkor redukált maradékrendszer $(\mathrm{mod}\ n)$, ha $lnko\,(b_i, n) = 1$, $b_i \not\equiv b_j\ (\mathrm{mod}\ n)$ és $B$ nem bővíthető további elemekkel ezen tulajdonságokkal. Az (egyetlen) ilyen $B$ halmaz jele $\mathbb{Z}_n^*$.
Használatosak még a $(\mathbb{Z}/n \cdot \mathbb{Z})^*$ és $(\mathbb{Z}/_{n\cdot\mathbb{Z}})^*$ jelölések is. □

**6.31. Definíció.** *Tetszőleges* $n \in \mathbb{Z}$ $(n \neq 0)$ számra jelölje $\varphi(n)$ a $(\mathrm{mod}\ n)$ redukált maradékosztályok számát, azaz legyen

$$\varphi\,(n) := |\mathbb{Z}_n^*|\,.$$

*Másképpen fogalmazva:* $\varphi\,(n)$ jelölje az $1$ és $n$ közötti, $n$-hez relatív prím számok számát.
A fenti függvényt **Euler-féle** $\varphi$ **függvény**nek nevezik (ld.még a 6.43 Definíciót). □

**6.32. Példa.** $\varphi\,(15) = 9$ mert $15$-höz relatív prím számok ($3$-mal és $5$-tel nem oszthatók): 1, 2, 4, 7, 8, 10, 11, 13, 14. Az 5.4. „Euler-féle $\varphi\,(n)$ függvény" alfejezetben megismerünk néhány képletet, amelyekkel $\varphi\,(n)$ értékét felsorolás nélkül is ki tudjuk számítani.

**Vigyázat:** $\mathbb{Z}_n^*$ nem zárt az összeadásra, erről bárki könnyen meggyőződhet. Szerencsére a szorzásra igen, ez éppen a 3.42. Állítás második fele.

A $(\mathbb{Z}_n^*, \cdot)$ struktúra tulajdonságait az alábbi Tételben foglaljuk össze:

**6.33. Tétel.** Tetszőleges $n \in \mathbb{Z}$ $(n \neq 0)$ *számra a* $\mathbb{Z}_n^*$ *redukált maradékrendszer zárt a szorzásra (azaz* $a, b \in \mathbb{Z}_n^*$ *esetén* $a \cdot b \in \mathbb{Z}_n^*$*), továbbá minden elemének van multiplikatív inverze, azaz az algebra nyelvén: a* $(\mathbb{Z}_n^*, \cdot)$ *struktúra kommutatív- (vagyis Abel-) csoport.*

**Bizonyítás.** A 3.42. Állítás második fele igazolja a zártságot.
A 6.36. Tétel és a 6.28. Tétel is igazolja, hogy minden elemnek van multiplikatív inverze.
■

**6.34. Megjegyzés.** Felhívjuk a figyelmet arra, hogy a 6.28. Tétel után következő Algoritmus nem csak az inverzelem létét állítja, hanem meg is keresi azt (mégpedig elég gyorsan). □

Végül megjegyezzük, hogy a csoportelméletben fontos $o(g)$ ("$g$ elem rendje") fogalmat az 5.7. "*Primitív gyökök és diszkrét logaritmus*" alfejezetben vezetjük be és vizsgáljuk.

## 6.3. Elsőfokú kongruencia-egyenletek

**6.35. Definíció.** Az

$$ax \equiv b\ \ (\mathrm{mod}\ m) \tag{6.4}$$

*kongruencia-egyenleteket* ($a, b, m$ adott, $x$ keresett) **elsőfokú** vagy **lineáris kongruenciá**nak nevezzük. □

A fenti (6.4)-nek **pontosan akkor** van megoldása, ha $ax = my + b$, azaz

$$ax - my = b, \tag{6.5}$$

vagyis (6.4)-t visszavezettük lineáris Diophantikus egyenletekre, amit Euklidesz algoritmusával könnyen meg tudunk oldani:

**6.36. Tétel.** *A (6.4) kongruencia-egyenletnek **pontosan akkor** van megoldása, ha*

$$lnko(a, m) \mid b\,. \tag{6.6}$$

*A megoldást az 5.4. Algoritmus megadja, a (6.5) átírás alapján.* □

A könyvhöz mellékelt EUKLDIO2D.EXE programot használhatjuk a (6.5) alakú egyenletek megoldására.

**6.37. Állítás.** *Tetszőleges $a, b, m \in \mathbb{Z}$ számokra a (6.4) egyenlet összes megoldása $(\mathrm{mod}\ m)$ a következő:*

$$x_i = x_0 + i \cdot \frac{m}{lnko\,(a, m)} \qquad i = 0, 1, \ldots, L-1$$

$$\textit{ahol} \quad L = lnko\,(a, m)\,,$$

*a megoldások száma $(\mathrm{mod}\ m)$ tehát* $L = lnko\,(a, m)$.

**Bizonyítás.** Az 5.10. Tételben szereplő (5.10) képlet alapján. ■

**6.38. Következmény.** *Ha $a$ és $m$ relatív prímek, akkor **bármely** $b \in \mathbb{Z}$ esetén* van *megoldása (6.4)-nek.* □

**6.39. Definíció.** Az

$$ax \equiv 1 \pmod{m} \tag{6.7}$$

kongruencia $x$ megoldását az $a \in \mathbb{Z}_m$ elem $(\mathrm{mod}\ m)$ **multiplikatív** (szorzási) **inverzé**nek nevezzük, és $a^{-1}\ (\mathrm{mod}\ m)$ -el jelöljük. □

**6.40. Tétel.** *Tetszőleges $a, m \in \mathbb{Z}$ számokra: $a$-nak pontosan akkor van multiplikatív inverze $(\mathrm{mod}\ m)$, ha $lnko\,(a, m) = 1$, azaz $a$ relatív prím $m$-hez képest, vagyis $a \in \mathbb{Z}_m^*$.* □

A fenti eredmények alapján például $\mathbb{Z}_m^*$ és $\mathbb{Z}_p$-ben minden elemnek van multiplikatív inverze, vagyis ezek a struktúrák csoportok. Továbbá:

**6.41. Megjegyzés.** Ha már megtaláltuk $a$-nak $a^{-1}$ multiplikatív inverzét $(\mathrm{mod}\ m)$, akkor *bármely* $b$ esetén a (6.4) kongruencia megoldása már csak egyetlen szorzásunkba kerül:

$$x \equiv b \cdot a^{-1} \pmod{m}\,. \tag{6.8}$$

Ez nagyméretű $a, m \in \mathbb{Z}$ számoknál lehet hasznos: egyedül csak a (6.7) kongruenciát kell megoldanunk!

A fenti (6.8) képlet lineáris algebrai megfelelője: ha már ismerjük az $A$ mátrix $A^{-1}$ inverzét, akkor utána *bármilyen* $\underline{b}$ esetén az $A \cdot \underline{x} = \underline{b}$ lineáris egyenletrendszer megoldása mindössze „csak” egy mátrix-vektor szorzás elvégzését igényli: $\underline{x} = A^{-1} \cdot \underline{b}$. □

**6.42. Példa. a)** oldja meg a $114x \equiv 3 \pmod{1683}$ egyenletet,
**b)** keresse meg $18$ multiplikatív inverzét $\pmod{175}$.

**Megoldás:** A maradékok szokás szerint $<\ldots>$ zárójelekben vannak (ld. 4.5. Jelölés):

**a)** a $114x - 1683y = 3$ lineáris Diophantikus egyenlet megoldása:
$< 114 > = < -1683 > \cdot 0 + < 114 >$,
$< -1683 > = < 114 > \cdot (-14) + < -87 >$,
$< 114 > = < -87 > \cdot (-1) + < 27 >$,
$< -87 > = < 27 > \cdot (-3) + < -6 >$,
$< 27 > = < -6 > \cdot (-4) + < 3 >$,
$< -6 > = < 3 > \cdot (-2) + < 0 >$,
ahonnan
$3 = lnko(114, -1683) = 1 \cdot < 27 > + 4 \cdot < -6 >$
$= 1 \cdot < 27 > + 4 \cdot (< -87 > - (-3) \cdot < 27 >) = 4 \cdot < -87 > + 13 \cdot < 27 >$
$= 4 \cdot < -87 > + 13 \cdot (< 114 > - (-1) \cdot < -87 >) = 13 \cdot < 114 > + 17 \cdot < -87 >$
$= 13 \cdot < 114 > + 17 \cdot (< -1683 > - (-14) \cdot < 114 >)$
$= 17 \cdot < -1683 > + 251 \cdot < 114 >$
$= 17 \cdot < -1683 > + 251 \cdot (< 114 > - 0 \cdot < -1683 >)$
$= 251 \cdot < 114 > + 17 \cdot < -1683 >$,
$x_0 = 251 \cdot C/d = 251, \quad y_0 = 17 \cdot C/d = 17$,
Az általános megoldás:
$x = x_0 + k \cdot b/d = 251 + k \cdot (-561),\ y = y_0 - k \cdot a/d = 17 - k \cdot 38 \quad (k \in \mathbb{Z})$.

A kongruencia megoldása: $x \equiv 251 \pmod{1683}$.

**b)** a $18x - 175y = 1$ lineáris Diophantikus egyenlet megoldása:
$< 18 > = < -175 > \cdot 0 + < 18 >$,
$< -175 > = < 18 > \cdot (-9) + < -13 >$,
$< 18 > = < -13 > \cdot (-1) + < 5 >$,
$< -13 > = < 5 > \cdot (-2) + < -3 >$,
$< 5 > = < -3 > \cdot (-1) + < 2 >$,
$< -3 > = < 2 > \cdot (-1) + < -1 >$,
$< 2 > = < -1 > \cdot (-2) + < 0 >$,
így $-1 = lnko(18, -175) = 1 \cdot < -3 > -(-1) \cdot < 2 >$
$= 1 \cdot < -3 > + 1 \cdot (< 5 > - (-1) \cdot < -3 >) = 1 \cdot < 5 > + 2 \cdot < -3 >$
$= 1 \cdot < 5 > + 2 \cdot (< -13 > - (-2) \cdot < 5 >) = 2 \cdot < -13 > + 5 \cdot < 5 >$
$= 2 \cdot < -13 > + 5 \cdot (< 18 > - (-1) \cdot < -13 >) = 5 \cdot < 18 > + 7 \cdot < -13 >$
$= 5 \cdot < 18 > + 7 \cdot (< -175 > - (-9) \cdot < 18 >) = 7 \cdot < -175 > + 68 \cdot < 18 >$
$= 7 \cdot < -175 > + 68 \cdot (< 18 > - 0 \cdot < -175 >) = 68 \cdot < 18 > + 7 \cdot < -175 >$
ahonnan
$x_0 = 68 \cdot C/d = -68, \quad y_0 = 7 \cdot C/d = -7$.
Az általános megoldás:
$x = x_0 + k \cdot b/d = -68 + k \cdot 175, \quad y = y0 - k \cdot a/d = -7 - k \cdot (-18) \ (k \in \mathbb{Z})$.

A fentiek alapján a $18$ multiplikatív inverze:

$$18^{-1} \equiv -68 \equiv 107 \pmod{175}.$$

Ellenőrzés: $18 \cdot 107 = 1926 \equiv 1 \pmod{175}$.

□

[SzI1]-ban sok kidolgozott példát találunk.

A mellékelt EUKLDIO2D.EXE programot is használhatjuk (6.5) alakú egyenletek megoldására.

(6.5) alakú egyenleteket sokszor kell megoldanunk számelméletben, például a 7.1. "*Kínai maradéktétel*" alfejezetben.

Az elsőfokú kongruenciák könnyű megoldhatóságával ellentétben a magasabbfokú kongruenciák (ld. 6.8. alfejezet) megoldására nincs gyors algoritmusunk, de éppen emiatt működik sok titkosírás (ld. 11. „*Bizonyítás 0 információval...*" fejezetben).

## 6.4. Euler-féle $\varphi(n)$ függvény

Már az előző fejezetekben is láttuk, hogy $\pmod{m}$ szorzások esetén az $m$-hez relatív prím számok kitüntetett szerepet játszanak, ez a későbbiekben is így lesz. Ezért megismételjük az alábbi fontos jelölést:

**6.43. Definíció.** Tetszőleges $n \in \mathbb{N}$ szám esetén $\varphi(n)$ jelölje az $n$-nél kisebb, $n$-hez relatív prím számok *számát*, azaz legyen

$$\varphi(n) := |\{a < n : lnko(a, n) = 1\}| .$$

A fenti függvényt **Euler-féle $\varphi$ függvénynek** nevezik. □

(*Leonhard Euler (1707-1783) svájci matematikus.*)

Angol nyelvű szakirodalomban néha találkozunk a "**totient-function**" (többszörös, lat.) elnevezéssel is.

**6.44. Megjegyzés.** A fenti definíció összhangban van a 6.31. Definícióval: $\varphi(n)$ éppen a $\mathbb{Z}_n^*$ redukált maradékrendszer mérete:

$$\varphi(n) = |\mathbb{Z}_n^*| . \quad \square \tag{6.9}$$

A 3.41. Állítás és a logikai szitaformula (részletesen ld. pl.[SzI2] 4. fejezet) segítségével $\varphi(n)$ „könnyen" kiszámítható:

**6.45. Állítás.** *Ha az $n \in \mathbb{N}$ szám törzstényezős felbontása $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ ahol $p_i \in \mathbb{P}$ páronként különböző prímszámok és $\alpha_i \geq 1$, akkor*

$$\varphi(n) = n - \sum_{i=1}^{r} \frac{n}{p_i} + \sum_{1 \leq i < j \leq r} \frac{n}{p_i \cdot p_j} - \sum_{1 \leq i < j < k \leq r} \frac{n}{p_i \cdot p_j \cdot p_k} + - \cdots$$
$$\cdots + (-1)^r \frac{n}{p_1 \cdot p_2 \cdot \cdots \cdot p_r} \tag{6.10}$$

□

A fenti képlettel mindössze „csak” két probléma van:

**(i)** Hány összeadandó tag is van a fenti összegben?

$$1 + r + \binom{r}{2} + \binom{r}{3} + \cdots + \binom{r}{r} = 2^r$$

a binomiális tétel szerint. Ez pedig *exponenciálisan sok* tag, ha $n$-nek sok különböző prímtényezője van. Ezt a problémát a következő 6.46. Tételben ki tudjuk küszöbölni.

**(ii)** A fenti (6.10) képlethez szükségünk van $n$ prímtényezős felbontására. Sajnos nem ismerünk még egyetlen olyan módszert sem, amely $n$ prímtényezős felbontása *nélkül* ki tudná számítani $\varphi(n)$ értékét. Ez vonatkozik az alfejezet alább következő képleteire is!

A fenti (6.10) kifejezés *könnyen* átírható rövidebb alakba:

**6.46. Tétel.** ***(Euler)*** *az előző állítás jelölései mellett*

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\ldots\left(1 - \frac{1}{pr}\right), \tag{6.11}$$

*vagy a $\prod$-jelölést használva*

$$\varphi(n) = n \cdot \prod_{i=1}^{r}\left(1 - \frac{1}{p_i}\right).$$

**Bizonyítás.** A (6.11) kifejezésben a zárójeleket beszorozva és rendezve éppen a (6.10) képletet kapjuk.

„Elegánsabb” megoldást kapunk a 6.50. Állítás és a 6.48. Példa felhasználásával. ■

**6.47. Megjegyzés.** Vegyük észre: csak az a lényeg, hogy az $n$ számot *mely* prímszámok osztják, de hogy pontosan melyik hatvánnyal az már nem. Sajnos éppen az $n$-et osztó számok (akár prím akár nem) megtalálása a *nehéz*, a kitevő kiszámítása már könnyű feladat lenne. □

**6.48. Példa.** Speciális esetek: ha $p, q \in \mathbb{P}$ különböző prímek, $t \in \mathbb{N}$ tetszőleges természetes szám, akkor

$\varphi(p) = p - 1,$

$\varphi(pq) = pq - p - q + 1 = (p-1)(q-1),$

$\varphi(p^t) = p^r - p^{r-1} = p^{r-1}(p-1).$ □

Nem csak elméletileg, hanem gyakorlatilag is hasznos a $\varphi$ függvény következő tulajdonsága:

**6.49. Definíció.** **(o)** Egy tetszőleges $\sigma : \mathbb{N} \to \mathbb{R}$ függvényt **számelméleti függvény**nek nevezünk.

**(i)** Egy tetszőleges $\sigma$ számelméleti függvény **gyengén multipilikatív**, ha *relatív prím* $m, n$ számokra teljesül

$$\sigma(m \cdot n) = \sigma(m) \cdot \sigma(n). \tag{6.12}$$

**(ii)** Egy tetszőleges $\sigma$ számelméleti függvény **teljesen / totálisan multiplikatív**, ha a fenti (6.12) összefüggés *tetszőleges* $m, n \in \mathbb{N}$ számokra teljesül. □

Sajnos a szakirodalom nem egységes abban, hogy a *multiplikatív* jelző egyedül mit jelentsen, tehát mindig tegyük mellé a *teljes* vagy *gyenge* melléknevet.

**6.50. Állítás.** *Az Euler-féle $\varphi$ függvény gyengén multiplikatív (de erősen nem).*

**Bizonyítás.** Legyen adott $m$ és $n$ egymáshoz képest relatív prím számok.
$\mathbb{Z}_m^*$ elemeit jelöljük $a_1, \ldots, a_{\varphi(m)}$ betűkkel, és tekintsük az

$$a_i + j \cdot m \qquad (1 \leq i \leq \varphi(m), 0 \leq j \leq n-1)$$

alakú számokat. Ezek mindegyike kisebb $m \cdot n$-nél, vagyis

$$a_i + j \cdot m \in \mathbb{Z}_{mn} \tag{6.13}$$

és $\mathbb{Z}_{mn}$-nek éppen ezek az elemei relatív prímek $m$-hez.

Ha kiválogatjuk közülük az $n$-hez relatív prímeket, akkor megkapjuk $\mathbb{Z}_{mn}^*$ elemeit, vagyis $\varphi(m \cdot n)$ értékét a 3.42. Állítás alapján.

Tetszőleges $i$-t rögzítve az $a_i + j \cdot m$ $(0 \leq j \leq n-1)$ számok teljes maradékrendszert alkotnak $(\mathrm{mod}\ n)$ a 6.18. Tétel szerint, amik közül nyilván $\varphi(n)$ db relatív prím $n$-hez.

Tehát a (6.13) számok közül valóban $\varphi(n) \cdot \varphi(m)$ ami relatív prím $mn$-hez, vagyis valóban

$$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n).$$

■

A [SzI1] Feladatgyűjteményben sok kidolgozott feladatot találunk $\varphi(n)$ kiszámítására és alkalmazására.

**6.51. Megjegyzés.** Milyen nagy is valójában $\varphi(n)$? A fenti képletek alapján

$$\frac{\varphi(n)}{n} = \prod_{i=1}^{r} \frac{p_i - 1}{p_i}$$

ami nagyon közel lehet 1-hez *ha* $n$-nek nincsenek kicsi prímosztói, azaz ebben az esetben

$$\varphi(n) \approx n$$

vagyis $\varphi(n)$ nagyon nagy. Ez a tény néhány későbbi algoritmusnál jól fog jönni. □

## 6.5. Maradékosztály-tételek

Ha az Olvasó nem tanult absztrakt algebrai struktúrákat, az első tételt át is ugorhatja.

**6.52. Tétel** (Lagrange)**.** *Ha $G$ egy véges csoport és $H$ egy részcsoportja $G$-nek, akkor $H$ elemeinek száma osztja $G$ elemeinek számát, azaz*

$$|H| \mid |G|. \quad \square$$

*(Joseph Louis Lagrange (1736-1813) francia matematikus.)*

Lagrange fenti általános algebrai tételét csak azért idéztük, mert bizonyítása egyszerű és rövid, és az alábbi 6.53. és 6.54. Tételek is egy tőmondatban következnek belőle. Vagyis nincs szükségünk a 6.53. és 6.54. Tételek közismert „nyakatekert" bizonyításaira.

Az alábbi tételeket a modern kódolások és titkosírások alapjainak is tekinthetjük:

**6.53. Tétel** (Euler „számelméleti" tétele)**.** *Ha* $m, a \in \mathbb{Z}$ *tetszőleges, relatív prím számok, akkor*

$$a^{\varphi(m)} \equiv 1 \pmod{m},$$

*ahol* $\varphi(m)$ *az Euler-féle "*$\varphi$*-függény" (ld. 6.43. Definíció).* □

Emlékeztetőül két speciális eset:

ha $p \in \mathbb{P}$ prímszám, akkor $\varphi(p) = p - 1$,

ha $m = p \cdot q \in \mathbb{Z}$ két prímszám szorzata, akkor $\varphi(pq) = (p-1) \cdot (q-1)$.

Így Euler fenti tételéből azonnal kapjuk Fermat alábbi „kis" tételét:

**6.54. Tétel** („kis" Fermat tétel)**.** *Ha* $p \in \mathbb{P}$ *prím és* $a \in \mathbb{Z}$, $p \nmid a$ *tetszőleges, akkor*

$$a^{p-1} \equiv 1 \pmod{p}. \quad \square \tag{6.14}$$

*(Pierre Fermat (1601-1664) francia jogász és matematikus.)*

Fermat fenti 6.54. Tétele alapján már *Bolyai János (1802-1860)* is próbálkozott nagy számok prím voltának eldöntésével, ezirányú vizsgálódásai során (ld. a 7.3. "*Álprímek*" alfejezetben) fedezte fel az alábbi Tételt:

**6.55. Tétel** (Bolyai János)**.** *Ha* $p$ *és* $q$ *egymástól különböző prímszámok,* $a$ *egy olyan egész szám, amely nem osztható sem* $p$*-vel, sem* $q$*-val, és amelyre* $a^{p-1} \equiv 1 \ (mod\ q)$ *és* $a^{q-1} \equiv 1 \pmod{p}$ *teljesül, akkor*

$$a^{pq-1} \equiv 1 \ (mod\ pq)$$

*is teljesül.* □

Ezzel a tétellel – amely csak 2000 körül került elő a kéziratokból – Bolyai János több mint 40 évvel megelőzte *Jeans (1877-1946) angol matematikust*, aki először közölte azt nyomtatásban (ld. Kiss Elemér [KE1] cikkében.)

*Bolyai Farkas(1775-1856)* és János ötletei alapján, a XX. században kifejlesztett meglepően hatékony algoritmust a 8.3. "*Álprímek*" alfejezetben ismertetjük.

Felhívjuk a figyelmet, hogy bármilyen nagy $m$ modulus és nagy $k$ kitevő esetén az $u^k$ hatványt $\pmod{m}$ könnyen és gyorsan ki tudjuk számolni (a 6.6 "*Nagy kitevőjű hatványozás*" alfejezetben), azonban a $\varphi(m)$ értékének kiszámítására nincs gyors algoritmus!

Euler fenti 6.53. és Fermat 6.54. Tételeinek általánosításai:

**6.56. Tétel.** ***(i)*** Tetszőleges $a \in \mathbb{Z}$ *egész és* $p \in \mathbb{P}$ *prímszámra*

$$a^p \equiv a \pmod{p}. \tag{6.15}$$

***(ii)*** *Tetszőleges* $m \in \mathbb{N}$ négyzetmentes *számra és* tetszőleges $a \in \mathbb{Z}$ *egész számra*

$$a^{\varphi(m)+1} \equiv a \pmod{m}. \tag{6.16}$$

(ii) egyetlen *nem-négyzetmentes* $m$ számra sem igaz, például
$3^{\varphi(9)+1} \equiv 0 \pmod{9}$ hiszen $\varphi(m) \geq 1$.

**Bizonyítás. (i)** nyilvánvaló: $p \nmid a$ esetén következik a kis Fermat-tételből, $p \mid a$ esetén pedig bármely $t \geq 1$ kitevőre $a^t \equiv 0 \equiv a \pmod{p}$.

**(ii)** Legyen $m = p_1 p_2 \dots p_k$ négyzetmentes. Tudjuk, hogy $\varphi(m) = \prod_{i=1}^{k} \varphi(p_i)$.
Először ismételjük meg az (i) gondolatmenetét $m$ minden $p_i$ prímosztójára: ha $p_i \mid a$ akkor bármely $t$ kitevőre $a^t \equiv 0 \equiv a \pmod{p_i}$. Ha pedig $p_i \nmid a$ akkor

$$a^{\varphi(m)+1} \equiv \left(a^{\varphi(p_i)}\right)^t \cdot a \equiv a \pmod{p_i} \tag{6.17}$$

a 6.54. „kis" Fermat-tétel szerint ($t$ lényegtelen).

Ha pedig (6.17) teljesül $m$ minden $p_i$ prímosztójára, akkor (6.16) is teljesül, a 6.14. Tétel alapján. ■

**6.57. Megjegyzés.** Bár a (6.14) egyenlőség (látszólag) több információt tartalmaz mint (6.15), valójában ekvivalensek: $a^p - a = a\,(a^{p-1} - 1)$ miatt $p \nmid a$ esetén nyilván $p \mid (a^{p-1} - 1)$. □

**6.58. Tétel** (Wilson)**.** $n \in \mathbb{N}$ *akkor és csak akkor prímszám, ha*

$$(n-1)! \equiv -1 \pmod{n}. \tag{6.18}$$

*vagy másképpen:*

$$n \mid (n-1)! + 1$$

*($n \neq 4$ összetett szám esetén nyilván $(n-1)! \equiv 0 \pmod{n}$.)*

*(John Wilson (1741-1793))*

**Bizonyítás.** $n = 2$ és $n = 3$ könnyen ellenőrizhető, tehát $n \geq 5$.
Legyen $n = p$ prím, ekkor a 6.40. Tétel szerint minden $1 \leq a \leq p-1$ számnak van multiplikatív inverze $1 \leq a^{-1} \leq p-1$, vagyis a $(p-1)!$ szorzatban az $a$ és $a^{-1}$ párok szorzata $\equiv +1 \pmod{p}$. Azonban $a_1 \equiv +1$ és $a_2 \equiv -1 \pmod{p}$ esetén (és csak erre a két számra) $a^{-1} \equiv a \pmod{p}$. Tehát

$$(p-1)! \equiv 1^{\frac{p-3}{2}} \cdot 1 \cdot (-1) \equiv -1 \pmod{p}.$$

Megfordítva: ha $n = a \cdot b$ összetett szám, akkor $a, b < n$ miatt $n \mid (n-1)!$, vagyis $n \nmid (n-1)! + 1$. ■

A fenti tétel megfordítása sajnos nagy számok prímtesztelésére azért nem jó, mert $(n-1)!$ bődülten nagy: ld. pl. a Stirling formulát a 2.6. Tételben.

## 6.6. Nagy kitevőjű hatványozás

Nem csak a titkosírások, hanem nagyon sok számelméleti vizsgálat egyik *alapművelete* a következő:

**6.59. Algoritmus. Nagy kitevőjű hatványozás (mod m):**
*Feladat:* Adottak a nagy (többszázjegyű) $u, k, m \in \mathbb{Z}$ számok. Meghatározandó az

$$x \equiv u^k \pmod{m}$$

értéke.

*Az algoritmus több lépésből áll, sok magyarázattal ellátva közöljük alább.*

A kb. $10^{100}$ *számjegyből* álló $u^k$ hatványt nyilván nem számoljuk ki teljesen. A 6.9 Tétel alapján nyilván minden részletszámítás végeredménye helyett annak $\pmod{m}$ *maradékát* vesszük, de $k \approx 10^{100}$ egymás utáni hatványozás is sokáig tartana.

Az alapvető ötlet a következő: ha már „elég nagy" kitevőjű hatványt kiszámoltunk, akkor annak pl. négyzetét (annak $\bmod m$ maradékát) egy szorzás megadja, és az már „sokkal nagyobb".

Vegyük tehát minden lépésben az előző lépésben kapott hatvány négyzetét: legyenek $u_0 :\equiv u$, $u_1 :\equiv u_0^2 \equiv u^2$, $u_2 :\equiv u_1^2 \equiv u^4, \ldots, u_i :\equiv u_{i-1}^2 \equiv u^{2^i}, \ldots \pmod{m}$. Láthatjuk, hogy a kitevők exponenciálisan gyorsan nőnek, vagyis a $t = \lceil \log_2 k \rceil \approx 100$-adik lépés után már a kitevő $2^t \geq k$. Igen, de ha $k$ nem $2^i$ alakú, akkor egyik $u_i$ sem megoldás.

Következő észrevételünk: 2 hatványainak összegeként minden szám előállítható, vagyis írjuk fel $k$-t kettes számrendszerben:

$$k = \overline{i_t\, i_{t-1}\, \ldots\, i_2\, i_1\, i_0}^{(2} = \sum_{j=0}^{t} i_j \cdot 2^j$$

ahol $i_j = 0$ vagy $= 1$ bináris jegyei $k$-nak. A hatványozás azonosságai miatt ekkor pedig

$$u^k = u^{\Sigma\, i_j \cdot 2^j} = \prod_{j=0,\ i_j \neq 0}^{t} u_{i_j}$$

vagyis: **azon $u_i$ számokat kell összeszoroznunk $\pmod{m}$ amely helyiértékein $k$ -nak $1$ áll**.

Megismételjük: az összes $u_{i_j}$ számot nem „hagyományosan" kell összeszoroznunk, hanem lépésenként, mindig csak a $\pmod{m}$ maradékot kiszámítva és figyelembe véve.

**6.60. Összegzés. Algoritmus vége.** □

A jegyzethez mellékelt HATVMODDD.EXE program segítségével tetszőleges (egyelőre csak legfeljebb $m < 2^{30} \approx 2 \cdot 10^9$) adatokkal gyakorolhatjuk $u^k \pmod{m}$ kiszámolását, az eredményeket file-ba is írathatjuk!

**6.61. Példa.** $6456^{4652} \pmod{9786}$ kiszámítása:

$k = 4652 = 1001000101100$ (bin) ,

$$\begin{aligned}
u[0] = u = 6456 = u^{2^0} &= 6456 \pmod{9786}\\
u[1] = (u[0])^2 = u^{2^1} &= 6456^2 \equiv 1362 \pmod{9786}\\
\mathbf{u[2]} = (u[1])^2 = u^{2^2} &= 1362^2 \equiv \mathbf{5490} \pmod{9786}\\
\mathbf{u[\,3]} = (u[2])^2 = u^{2^3} &= 5490^2 \equiv \mathbf{9006} \pmod{9786}\\
u[4] = (u[3])^2 = u^{2^4} &= 9006^2 \equiv 1668 \pmod{9786}\\
\mathbf{u[\,5]} = (u[4])^2 = u^{2^5} &= 1668^2 \equiv \mathbf{3000} \pmod{9786}\\
u[6] = (u[5])^2 = u^{2^6} &= 3000^2 \equiv 6666 \pmod{9786}\\
u[7] = (u[6])^2 = u^{2^7} &= 6666^2 \equiv 7116 \pmod{9786}\\
u[8] = (u[7])^2 = u^{2^8} &= 7116^2 \equiv 4692 \pmod{9786}\\
\mathbf{u[\,9]} = (u[8])^2 = u^{2^9} &= 4692^2 \equiv \mathbf{6150} \pmod{9786}\\
u[10] = (u[9])^2 = u^{2^{10}} &= 6150^2 \equiv 9396 \pmod{9786}\\
u[11] = (u[10])^2 = u^{2^{11}} &= 9396^2 \equiv 5310 \pmod{9786}\\
\mathbf{u[12]} = (u[11])^2 = u^{2^{12}} &= 5310^2 \equiv \mathbf{2634} \pmod{9786}
\end{aligned}$$

így

$u^k \equiv 5490 \cdot 9006 \cdot 3000 \cdot 6150 \cdot 2634 \equiv 4068 \cdot 3000 \cdot 6150 \cdot 2634 \equiv 858 \cdot 6150 \cdot 2634 \equiv$
$\equiv 2046 \cdot 2634 \equiv 6864 \pmod{9786}$,

vagyis $6456^{4652} \equiv 6864 \pmod{9786}$. □

[SzI1] 46. ill. 109-112. oldalain több kidolgozott példát találunk a fenti algoritmusra.

**6.62. Megjegyzés.** **(i)** Az algoritmus lépésigénye legfeljebb $2t = 2\lceil \log_2 k \rceil$ szorzás $(\mathrm{mod}\ m)$, vagyis $k$ számjegyeinek számának mindössze kétszerese, vagyis ez egy *lineáris* algoritmus! Számítógép-memóriát szinte semmit nem használunk fel: lényegében az adatokat és $k$ számjegyeit ($t$ db bit) kell tárolnunk!

**(ii)** $k \geq m$ esetén az $u^i$ sorozat (nyilván) *mindenképpen* ciklikus, így a hatványkitevőt csökkenthetnénk ha ismernénk a ciklus hosszát – ami viszont nem egyszerű feladat.

Ha pedig $u$ és $m$ relatív prímek, akkor még az is feltehető, hogy a $k$ kitevő $\varphi(m)$-nél is kisebb, ugyanis ekkor $k \geq \varphi(m)$ esetén a 6.53. Euler Tétel miatt

$$u^k \equiv u^\kappa \pmod{m} \quad \text{ahol} \quad \kappa \equiv k \pmod{\varphi(m)}$$

hiszen $k = y \cdot \varphi(m) + \ell$ esetén

$$u^k \equiv \left(u^{\varphi(m)}\right)^y \cdot u^\ell \equiv 1 \cdot u^\ell \pmod{m}.$$

Sajnos $\varphi(m)$ értékét nem olyan egyszerű meghatározni, mint tudjuk. □

## 6.7. Primitív gyökök és diszkrét logaritmus

Mivel ebben az alfejezetben egy rögzített $\mathbb{Z}_m^*$ halmazban dolgozunk ($m \in \mathbb{Z}$ vagy $m = p \in \mathbb{P}$), ezért a $\equiv$ jel után nem írjuk ki mindig a $(\mathrm{mod}\ m)$ illetve a $(\mathrm{mod}\ p)$ jelölést.

$\mathbb{Z}_m^*$-ben (vagy akár egy tetszőleges véges csoportban) egy tetszőleges elemet önmagával szorozgatva (hatványozva) érdekes és hasznos jelenségekkel találkozunk.

**6.63. Megjegyzés. (i)** Azért szorítkozunk csak $\mathbb{Z}_m^*$ halmazra, mert a $\mathbb{Z}_m$ halmazban ($m$ összetett szám esetén) kétféle elemek vannak: $m$-hez relatív prím, és $m$-mel valamilyen közös osztóval rendelkező elemek. Márpedig ez utóbbiak halmaza nem zárt a szorzásra.
(ii) Az alábbi eredmények és bizonyításuk tetszőleges véges $G$ csoportokban is igazak. □

**6.64. Állítás.** ***(i)*** *Legyen $a \in \mathbb{Z}_m^*$ tetszőleges elem. Ekkor egyértelműen van olyan ($a$-tól és $m$-től függő),* legkisebb $d < m$ *kitevő, amelyre*

$$a^d \equiv 1.$$

*Erre a $d$ kitevőre érvényes továbbá, hogy $a^d \equiv a^{d+k}$, sőt*

$$a^i \equiv a^j \iff i \equiv j \ (\mathrm{mod}\ d) \tag{6.19}$$

*tetszőleges $i, j, k \in \mathbb{N}$ kitevőkre.*

***(ii)*** *Ha $a \in \mathbb{Z}_m^*$ olyan maradék, amelyhez tartozó $d = \varphi(m) = |\mathbb{Z}_m^*|$, akkor a hatványai (nem sorrendben) kiadják $\mathbb{Z}_m^*$ elemeit, azaz*

$$\left\{1, a, a^2, \ldots, a^{d-1}\right\} = \mathbb{Z}_m^* \tag{6.20}$$

*(ne feledjük: $a^1 = a$ és $a^d \equiv 1 = a^0$).*

***Bizonyítás. (i)*** *$a = 1$ esetén $d = 1$, tehát $1 < a$. Mivel az alaphalmaz ($\mathbb{Z}_m^*$) véges, ezért előbb-utóbb ismétlődést kapunk: $a^e \equiv a^f$ valamilyen $e < f$ természetes számokra. Ekkor nyilván $a^{f-e} = a^h \equiv 1$. Nyilvánvalóan $d$ éppen a legkisebb ilyen $h$ kitevő. Az (i) állítás további része nyilvánvaló.*
***(ii)*** *Ha $d$ minimális, akkor $a$ hatványai $0$-tól $d-1$-ig mind különbözőek. Ekkor pedig (6.20) nyilvánvaló, hiszen a két halmaznak ugyanannyi eleme van.* ■

**6.65. Definíció. (i)** Tetszőleges $a \in \mathbb{Z}_m^*$ elemre az $a$ **elem rendje**, $o(a)$ jelölje a 6.64. Állítás (i) pontjában szereplő *(legkisebb)* $d$ kitevőt.
Az $a$ elemet szokás $d$-edik **egységgyök**nek is hívni, hiszen $a^d \equiv 1$.
**(ii)** Az $a \in \mathbb{Z}_m^*$ elemet **primitív egységgyök**nek, röviden csak **primitív gyök**nek, vagy (multiplikatív) **generátor elem**nek hívjuk, ha $o(a) := \varphi(m)$. Másképpen: ha (6.20), vagyis a 6.64. Állítás (ii) pontja teljesül.
Ebben az esetben a $(\mathbb{Z}_m^*, \odot)$ csoportot **ciklikus**nak nevezzük.
A primitív gyököket általában $g$ betűvel jelöljük. □

A hatványozás és a logaritmus jól ismert azonosságai $\mathbb{Z}_m^*$-ban is érvényben vannak. Egyedül az alábbi összefüggés érdemel említést:

**6.66. Állítás.** $o(ab) = lkkt(o(a), o(b))$ *tetszőleges* $a, b \in \mathbb{Z}_m^*$ *elemekre.* □

De elfelejtettük a legfontosabbat: mely $m \in \mathbb{N}$ modulusra *létezik* is primitív gyök? Az alábbi eredményeket nem bizonyítjuk, felsőfokú *algebra* vagy számelmélet könyvekben megtalálható.

**6.67. Tétel.** $\mathbb{Z}_m^*$*-ban* pontosan akkor *van primitív gyök, ha* $m = 2$, $m = 4$, $m = p^a$ *vagy* $m = 2p^a$ *ahol* $p \in \mathbb{P}$ páratlan *prímszám és* $\alpha \geq 1$ *tetszőleges természetes szám.* □

**6.68. Állítás.** $\mathbb{Z}_p^*$*-ben pontosan azon* $d$ *kitevőkre létezik* $d$*-edrendű elem* $a \in \mathbb{Z}_p^*$ *(azaz* $o(a) = d$*), melyekre* $d \mid p-1$. □

**6.69. Megjegyzés.** $\mathbb{Z}_{p^\alpha}^*$ nem tévesztendő össze $GF(p^\alpha)$-vel (= $p^\alpha$-ad rendű véges test =Galois Field, ld. Függelék) . □

Kis $m$ értékekre könnyen kereshetünk primitív gyököket, sőt hatvány- és logaritmus táblázatokat is könyen készíthetünk, amiket például könyvünk végén is találhatunk.

**6.70. Definíció.** Legyen $g \in \mathbb{Z}_m^*$ egy (tetszőleges) primitív gyök, $a \in \mathbb{Z}_m^*$ tetszőleges szám és $a = g^k$ ahol $k < \varphi(m)$. Ezt az egyértelmű $k$-t $a$-nak ($g$-alapú) **diszkrét logaritmusá**nak vagy (számelméleti) **index**ének nevezzük, és a $k = \log_g(a)$ vagy $k = ind_g(a)$ jelölést használjuk. A jelölésben ugyan $(\mathrm{mod}\ m)$ nem szerepel, de a szövegkörnyezetben mindig megemlítjük. □

Az elkészített indextáblázatok haszna nagy $m$ modulusoknál nyilvánvaló (bár az alábbi példában csak kis számokkal mutatjuk meg a táblázatok használatát):

**6.71. Példa.** Legyen $p = 47$, $g = 5$, a könyv végén (116. oldal) levő *jobboldali* táblázatban vannak $g$ hatványai, tehát például $g^{12} \equiv 18 \pmod{47}$.
Továbbá $a = 26$ és $b = 37$ esetén, a *baloldali* táblázat szerint
$\log_g(a) = \log_5(26) = 29$ és $\log_g(b) = \log_5(37) = 42$.

Az $ab$ szorzást $(\mathrm{mod}\ p)$ megint összeadásra vezetjük vissza $(\mathrm{mod}\ p-1)$:

$\log_g(ab) = \log_g(a) + \log_g(b) = 29 + 42 \equiv 25 \pmod{46}$

tehát $ab = 26 \cdot 37 \equiv g^{25} = 22 \pmod{47}$ (tessék ellenőrizni!)

(a jobboldali táblázat szerint – mint a hagyományos logaritmus esetén).

Továbbá, például az $x^2 \equiv a = 26$ és $y^2 \equiv b = 37 \pmod{47}$ egyenleteket is könnyen megoldhatjuk:

$2 \cdot \log_5(x) \equiv \log_g(a) = 29 \pmod{46}$ miatt ilyen $x$ nem létezik,

míg a

$2 \cdot \log_5(y) \equiv \log_g(b) = 42 \pmod{46}$

kongruencia megoldásai:

$\log_5(y_1) \equiv 21 \pmod{46} \implies y_1 \equiv g^{21} \equiv 15 \pmod{47}$,

$\log_5(y_2) \equiv 44 \pmod{46} \implies y_2 \equiv g^{44} \equiv 32 \equiv -15 \pmod{47}$

(tessék ellenőrizni – és a következő „Magasabbfokú kongruenciák” alfejezet nehézségeivel összevetni!). □

Már csak primitív gyököket kell keresnünk és a táblázatokat elkészítenünk adott (nagyméretű) $m$ modulusokra.

Azonban sem képletünk sem gyors algoritmusunk nincs primitív gyökök keresésére. Euler 6.53. Tételéből következik, hogy bármely $m$-hez relatív prím $a$ számra

$$a^{\frac{\varphi(m)}{2}} \equiv \pm 1 \pmod{m} \tag{6.21}$$

tehát nyilvánvaló a következő:

**6.72. Állítás.** *Tetszőleges $m, g \in \mathbb{Z}$ relatív prím számokra $g$* pontosan akkor *primitív gyök* $\pmod{m}$ *ha*

$$g^{\frac{\varphi(m)}{2}} \equiv -1 \pmod{m}. \quad \square \tag{6.22}$$

A fenti (6.22) feltételt ugyan könnyen és gyorsan ellenőrizhetjük a 6.6. „*Nagy kitevőjű hatványozás*" alfejezetben megismert algoritmussal, *de:* melyik $a \in \mathbb{Z}_m^*$ számot vizsgáljuk? Esetleg: $\mathbb{Z}_m^*$ elemeinek hányadrésze primitív gyök?

Előrebocsájtjuk, hogy (többszázjegyű $m$ modulusoknál) még ha találtunk is egy $g$ primitív gyököt, az indextáblázatot már *mérete* miatt sem tudjuk felírni, továbbá arra sincs gyors algoritmusunk, hogy egy tetszőleges $a \in \mathbb{Z}_m^*$ szám $g$ alapú logaritmusát meghatározzuk.

Az alábbi eredmények ismeretesek:

**6.73. Tétel.** *Ha $\mathbb{Z}_m^*$-ban van primitív gyök, akkor* pontosan $\varphi(\varphi(m))$ *db primitív gyök van $\mathbb{Z}_m^*$-ban.* $\square$

**6.74. Következmény.** *$\mathbb{Z}_m^*$-ban véletlenszerűen kiválasztva egy számot, az*

$$\frac{\varphi(\varphi(m))}{\varphi(m)}$$

*valószínűséggel lesz primitív gyök.*

*Speciálisan, ha $m = p \in \mathbb{P}$ prímszám, akkor ez a valószínűség* $\dfrac{\varphi(p-1)}{p-1}$. $\square$

**6.75. Tétel.** Van *prímszámok olyan $(p_n)_{n=1}^{\infty}$ végtelen sorozata, amelyre a fenti valószínűség* 0*-hoz tart:*

$$\frac{\varphi(p_n - 1)}{p_n - 1} \to 0. \qquad \square$$

**6.76. Tétel** (Burgess, 1962)**.** *Bármilyen $C, \varepsilon \in \mathbb{R}^+$ konstansokra létezik $Cp^{1/4+\varepsilon}$-nél kisebb primitív gyök* $\pmod{p}$*, ha $p$ elég nagy prímszám.* $\square$

## 6.8. Magasabbfokú kongruenciák

Az alábbi egyenletek sem csak elméletileg érdekesek, hanem sok algoritmusban is fontosak (prímtesztek, bizonyítás 0 információval, . . . ).

**6.77. Definíció. Magasabbrendű kongruenciá**nak nevezzük az

$$x^k \equiv a \pmod{m} \tag{6.23}$$

alakú egyenleteket ha $k \geq 2$.

Speciálisan, $k = 2$ esetén az

$$x^2 \equiv a \pmod{m} \tag{6.24}$$

kongruenciát **négyzetes** vagy **kvadratikus kongruenciá**nak nevezzük, míg azokat az $a \in \mathbb{Z}_m$ maradékokat, amelyekre (6.24)-nek van megoldása, **négyzetes** vagy **kvadratikus maradékok**nak nevezzük $\pmod{m}$.

$x$-et az $a$ szám **k-adik gyökének**, illetve **négyzetgyökének** is nevezzük $\pmod{m}$. □

Tehát *nem* a (6.24) egyenlet megoldásait hívjuk négyzetes maradékoknak.
Egyszerűség végett a továbbiakban csak az $a \neq 0$ számokkal foglalkozunk.

Természetesen merül fel a következő probléma:

**6.78. Probléma.** *Milyen $m, a \in \mathbb{Z}$ és $k \in \mathbb{N}$ esetén van megoldása a 6.23 kongruenciának?* □

Először vizsgáljuk meg a négyzetes maradékok problémáját az $m = p \in \mathbb{P}$ prímmodulus esetben, majd később foglalkozunk tetszőleges $m \in \mathbb{Z}$ modulussal.

**6.79. Állítás.** *$k = 2$ esetén:*

***(i)** Tetszőleges $p \in \mathbb{P}$ prím, $p > 2$ és $a \in \mathbb{Z}_p$ számokra: ha a (6.24) kongruenciának van megoldása, akkor **pontosan kettő** megoldása van.*

***(ii)** Minden $p \in \mathbb{P}$ prímszámra $\mathbb{Z}_p \setminus \{0\}$ elemeinek* pontosan *a fele ($= \frac{p-1}{2}$) négyzetes- és ugyanennyi nemnégyzetes- maradék.*

***Bizonyítás. (i)** $(p-x)^2 \equiv x^2 \pmod{p}$ és $p$ páratlan esetén $p - x \not\equiv x \pmod{p}$ miatt legalább két gyök van.*
*Ha pedig $x^2 \equiv y^2 \pmod{p}$ akkor $x^2 - y^2 = (x-y)(x+y) \equiv 0 \pmod{p}$, és $p$ prímtulajdonsága miatt $p \mid (x-y)$ vagy $p \mid (x+y)$, azaz valóban $x \equiv \pm y \pmod{p}$.*

***(ii)** Az előző gondolatmenet szerint az $1, 2, \ldots, \frac{p-1}{2}$ számok négyzetei mind különbözőek és kiadják az összes négyzetes maradékokat.* ■

**6.80. Állítás.** Tetszőleges $m \in \mathbb{Z}$ páratlan *és $a \in \mathbb{Z}_m$ számokra $k = 2$ esetén ha a (6.24) kongruenciának van megoldása, akkor **legalább kettő** megoldása van.*

***Bizonyítás.** Az előző állítás bizonyításának első sora tetszőleges páratlan $m$ modulusra is jó.* ■

A következő tétel bizonyítása már nem olyan egyszerű mint a fentieké.

**6.81. Tétel** ([KN],II.2.)**.** *Ha $m \in \mathbb{Z}$ tetszőleges páratlan szám, akkor $\mathbb{Z}_m^*$ elemeinek legalább a fele négyzetes nemmaradék.* □

Hiába könnyű elvileg felsorolni a négyzetes maradékokat (csak $i^2$ maradékait kellene egy táblázatban beikszelnünk), de nagy $p$ modulusra nem érnénk a végére. Még akkor sem, ha tudnánk egy $g \in \mathbb{Z}_p$ primitív gyököt, aminek ugye pontosan a páros kitevőjű hatványai a négyzetes maradékok.

A következő tétel segítségével könnyen és *gyorsan* eldönthetjük (még nagy $p \in \mathbb{P}$ modulusokra is), hogy mely $a$ számok a négyzetes maradékok:

**6.82. Tétel** (Euler-lemma)**.**

*Tetszőleges $p \in \mathbb{P}$ prím modulusra egy $p$-hez relatív prím $a$ szám*
*(vagyis $p \nmid a$ )* akkor és csak akkor *négyzetes maradék, ha*

$$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}. \tag{6.25}$$

**Bizonyítás.** A kis Femat-tétel alapján bármely $p$-hez relatív prím $a$ számra $\left(a^{\frac{p-1}{2}}\right)^2 \equiv 1 \pmod{p}$ tehát

$$a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}.$$

Legyen $g \in \mathbb{Z}_p^*$ primitív gyök a 6.67. Tétel szerint és legyen $x \equiv g^i$, $a \equiv g^j \pmod{p}$ ahol $i, j < p$.
A (6.24) kongruencia ekvivalens az

$$g^{2i} \equiv g^j \pmod{p}$$

azaz

$$2i \equiv j \pmod{p-1} \tag{6.26}$$

kongruenciával hiszen $o(g) = p - 1$.
Mivel $p - 1$ páros, ezért (6.26) szerint $j$ mindenképpen páros ha $a$ négyzetes maradék.

Ha $j = 2i'$ páros akkor egyrészt

$$a \equiv g^{2i'} \equiv \left(g^{i'}\right)^2 \pmod{p}$$

négyzetes maradék, másrészt

$$a^{\frac{p-1}{2}} \equiv \left(g^{2i'}\right)^{\frac{p-1}{2}} \equiv \left(g^{i'}\right)^{p-1} \equiv 1 \pmod{p}$$

a 6.54. „kis” Femat Tétel alapján.

Ha $j = 2i' + 1$ páratlan, akkor a fentiek szerint $a$ nem négyzetes maradék, másrészt

$$a^{\frac{p-1}{2}} \equiv \left(g^{2i'+1}\right)^{\frac{p-1}{2}} \equiv \left(g^{i'}\right)^{p-1} \cdot g^{\frac{p-1}{2}} \equiv -1 \pmod{p}$$

hiszen $g$ rendje $o(g) = p - 1$ miatt $g^{\frac{p-1}{2}} \not\equiv 1 \pmod{p}$, vagyis

$$g^{\frac{p-1}{2}} \equiv -1 \pmod{p}.$$

Bebizonyítottuk az Euler-lemmát. ■

A fenti eredmények ismeretében már kicsit világosabban látjuk a 6.78 problémát:

**6.83. Megjegyzés.** Emlékeztetünk rá, hogy a 6.6. "*Nagy kitevőjű hatványozás*" alfejezetben megismert módszerrel a (6.25) feltétel könnyen (és gyorsan) ellenőrizhető.

Arra is felhívjuk a figyelmet, hogy a fenti Tétel ill. (6.25) algoritmus csak (6.24) megoldhatóságát (a négyzetgyök létezését) dönti el, de *nem adja meg* $a$ *négyzetgyökét* – nem oldja meg a (6.24) kongruenciát! Sajnos a négyzetgyökök tényleges megtalálására *nincs* gyors algoritmusunk, lényegében csak egy teljes indextáblázat segíthetne (ld. a 6.71. Példát), de annak már mérete is exponenciálisan nagy. Sebaj, épp emiatt lehet különböző titkosírásokra használni a négyzetes maradékokat!

Bár a fenti Euler-lemma segítségével bármely, akármilyen nagyméretű $a$ és $p \in \mathbb{P}$ számokra könnyen és gyorsan kiszámíthatjuk (számítógéppel), hogy $a$ négyzetes maradék-e $(\mathrm{mod}\ p)$ vagy sem, kézi számolásokhoz és a probléma elméleti vizsgálatára sokszor az alábbi eredmények hasznosabbak (bizonyításuk megtalálható például [SA1] vagy [KN]-ban):

**6.84. Definíció.** Tetszőleges $p \in \mathbb{P}$, $p > 2$ prím és $a \in \mathbb{Z}_p$ számra az $\left(\frac{a}{p}\right)$ **Legendre-szimbólum** a következő:

$$\left(\frac{a}{p}\right) := \begin{cases} 0 & \text{ha } p \mid a \\ 1 & \text{ha } a \text{ négyzetes maradék } (\mathrm{mod}\ m) \\ -1 & \text{ha } a \text{ nem négyzetes maradék } (\mathrm{mod}\ m)\,. \end{cases} \quad \square$$

*(Adrien Marie Legendre (1752-1833), francia matematikus.)*

**6.85. Definíció.** Tetszőleges $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_t^{\alpha_t} \in \mathbb{N}$ és $a \in \mathbb{Z}_m$ számra az $\left(\frac{a}{m}\right)$ **Jacobi-szimbólum** a következő:

$$\left(\frac{a}{m}\right) := \left(\frac{a}{p_1}\right)^{\alpha_1} \left(\frac{a}{p_2}\right)^{\alpha_2} \dots \left(\frac{a}{p_t}\right)^{\alpha_t}. \quad \square$$

*(Carl Gustav Jakob Jacobi (1804-1851), német matematikus.)*

**6.86. Állítás.** *A 6.82. Tétel szerint tetszőleges $p \in \mathbb{P}$ prím és $a \in \mathbb{Z}$ egész számokra*

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}. \quad \square$$

**6.87. Megjegyzés.** Felhívjuk a figyelmet arra, hogy a Legendre szimbólum esetén

$$\left(\frac{a}{p}\right) = 1 \quad \Longleftrightarrow \quad a \text{ négyzetes maradék } (\mathrm{mod}\ p)$$

de a Jacobi szimbólumnál $\left(\frac{a}{m}\right) = 1$ egyáltalában *nem jelenti azt*, hogy $a$ négyzetes maradék lenne $(\mathrm{mod}\ m)$, például $\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right) \cdot \left(\frac{2}{5}\right) = 1$ de 2 *nem* négyzetes maradék $(\mathrm{mod}\ 15)$. $\square$

**6.88. Tétel.** ***(i)*** *ha* $a_1 \equiv a_2 \pmod{p}$ *akkor* $\left(\frac{a_1}{p}\right) = \left(\frac{a_2}{p}\right)$,

***(ii)*** *tetszőleges* $a, b \in \mathbb{Z}$ *számokra* $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$,

***(iii)*** *ha* $p \nmid b$ *akkor* $\left(\frac{ab^2}{p}\right) = \left(\frac{a}{p}\right)$,

***(iv)*** $\left(\frac{1}{p}\right) = 1$,

$$\left(\frac{-1}{p}\right) = \begin{cases} 1 & \textit{ha} \quad p \equiv 1 \pmod{4} \quad \textit{vagy } p = 2 \\ -1 & \textit{máskor} \end{cases},$$

***(v)*** $$\left(\frac{2}{p}\right) = \begin{cases} 1 & \textit{ha} \quad p \equiv \pm 1 \pmod{8} \\ -1 & \textit{máskor} \end{cases},$$

***(vi)*** $$\left(\frac{-2}{p}\right) = \begin{cases} 1 & \textit{ha} \quad p \equiv 1 \textit{ vagy } \equiv 3 \pmod{8} \\ -1 & \textit{máskor} \end{cases}. \quad \square$$

**6.89. Tétel** (Kvadratikus reciprocitás = Négyzetes megfordítás)**.** *Tetszőleges* $m, n \in \mathbb{Z}$ páratlan *számokra*

$$\left(\frac{m}{n}\right) = (-1)^{\frac{(m-1)(n-1)}{4}} \cdot \left(\frac{n}{m}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \textit{ha } n \equiv m \equiv 3 \pmod{4} \\ \left(\frac{n}{m}\right) & \textit{máskor} \end{cases}$$

$\square$

**6.90. Tétel.** *Tetszőleges* $m \in \mathbb{Z}$ páratlan *számra*

$$\left(\frac{2}{m}\right) = (-1)^{\frac{n^2-1}{8}} = \begin{cases} 1 & \textit{ha} \quad n^2 \equiv 1 \pmod{16} \\ -1 & \textit{máskor} \end{cases}. \quad \square$$

**6.91. Példa.** $\left(\frac{7411}{9283}\right) = -\left(\frac{9283}{7411}\right) = -\left(\frac{1872}{7411}\right) = -\left(\frac{2^4}{7411}\right)\left(\frac{117}{7411}\right) =$
$-1 \cdot \left(\frac{7411}{117}\right) = -\left(\frac{40}{117}\right) = -\left(\frac{2^4 \cdot 2}{117}\right)\left(\frac{5}{117}\right) = \left(\frac{5}{117}\right) = \left(\frac{117}{5}\right) = \left(\frac{2}{5}\right) = 1. \quad \square$

Az alábbi algoritmus sajnos nem tökéletes: előtte kell valahonnan egy négyzetes *nemmaradék*ot keresnünk $\bmod p$.

**6.92. Algoritmus.** ([KN],II.2.)
Tetszőleges $p \in \mathbb{P}$ prím modulusra és egy tetszőleges $a \in \mathbb{Z}_p$ négyzetes maradékra ($p \nmid a$) gyors algoritmust adunk $a$ négyzetgyökeinek megtalálására – *feltéve, hogy már ismert egy* $n \in \mathbb{Z}_p$ *nemmaradék.* (6.79. Állítás szerint egy véletlenül választott $n \in \mathbb{Z}_p$ szám pontosan 50% eséllyel nemmaradék $(\mathrm{mod}\ p)$.)
Az algoritmusban végig csak a $\equiv$ jelet fogjuk kiírni, a $(\mathrm{mod}\ p)$ jelölést elhagyjuk.

Tehát $n$ egy nemmaradék és $a$-nak keressük a négyzetgyökét.
Legyen $p-1 = 2^\beta \cdot s$ ahol $s$ páratlan és legyen $r :\equiv a^{(s+1)/2}$. Ekkor

$$a^{s\cdot 2^{\beta-1}} \equiv a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) = 1$$

vagyis $a^{-1}r^2 \in \mathbb{Z}_p$ egy $2^{\beta-1}$-dik egységgyök.

Ötlet: $r$-et kicsit módosítva fogunk olyan $x$-et kapni, amelyre $a^{-1}x^2 \equiv 1$.

Legyen $b :\equiv n^s$ ekkor kiszámolható, hogy a $\left\{b, b^2, b^3, \ldots, b^{2^\beta}\right\}$ elemek mind különbözőek $(\mathrm{mod}\ p)$ és $b^{2^\beta} \equiv 1$.

Keresünk egy olyan $j < 2^{\beta-1}$ kitevőt, amelyre $x = b^j \cdot r$ négyzetgyöke lesz $a$-nak. $j$-nek bináris számjegyeit fogjuk meghatározni:

$$j = \overline{j_{\beta-2},\ j_{\beta-3},\ \ldots, j_2,\ j_1,\ j_0}^{(2} = \sum_{i=0}^{\beta-2} 2^i j_i.$$

**1.** Tudjuk, hogy $\left(r^2 \cdot a^{-1}\right)^{2^{\beta-1}} \equiv \pm 1$. Így $\equiv 1$ esetén legyen $j_0 := 0$; $\equiv -1$ esetén legyen $j_0 := 1$.
**2.** Ha már a $j_0, j_1, \ldots, j_{k-1}$ jegyeket már megkaptuk ($k \le \beta - 2$), akkor kiszámolható, hogy $\left((b^{J_{k-1}} \cdot r)^2 \cdot a^{-1}\right)^{2^{\beta-k-1}} \equiv 1$, ahol $J_{k-1} := \sum_{i=0}^{k-1} 2^i j_i$, vagyis $\left((b^{J_{k-1}} \cdot r)^2 \cdot a^{-1}\right)^{2^{\beta-k-2}} \equiv \pm 1$.
Ennek megfelelően legyen $j_k := 0$ vagy $j_k := 1$ mint $j_0$ esetében.

Ekkor

$$x :\equiv b^j \cdot r$$

egy négyzetgyöke $a$-nak.
Vége az algoritmusnak. □

**Az algoritmus futásideje** $\mathcal{O}\left(\log^4(p)\right)$. □

Mégegyszer hangsúlyozzuk, hogy nincs tudomásunk olyan *polinomiális* algoritmusról, ami biztosan adna egy négyzetes nemmradékot. A 6.79. Állítás alapján csak annyit tudunk biztosan, hogy véletlenszerűen választva $n$ különböző számot $\mathbb{Z}_p$-ben, legalább $1 - 2^{-n}$ valószínűséggel van a választott számok között legalább egy négyzetes nemmaradék.

**6.93. Példa.** A fenti algoritmussal keressük meg $a = 186$ egy négyzetgyökét $(\mathrm{mod}\ 401)$.

$n = 3$ nemmaradék mert $3^{400/2} \equiv -1 \pmod{401}$.
$p - 1 = 400 = 2^4 \cdot 25$ ($\beta = 4$) miatt $r \equiv a^{(s+1)/2} \equiv 186^{26/3} \equiv 103$ és $b \equiv n^s \equiv 3^{25} \equiv 268$, továbbá $a^{-1} \equiv 235$ és $r^2 \cdot a^{-1} \equiv 98$ (ami állítólag $2^{\beta-1} = 8$-adik egységgyök: $98^8 \equiv 1$).

Most keressük meg $j$ számjegyeit:

$k=0$: $\left(r^2\cdot a^{-1}\right)^{2^{\beta-1}} = 98^8 \equiv -1$ miatt $j_0 = 1$.

$k=1$: $\left(\left(b^1\cdot r\right)^2\cdot a^{-1}\right)^{2^{\beta-k-2}} \equiv \left((268\cdot 103)^2\cdot 235\right)^2 \equiv 1 \Longrightarrow j_1 = 0$,

$k=2$: $\left(\left(b^1\cdot r\right)^2\cdot a^{-1}\right)^{2^{\beta-k-3}} \equiv (268\cdot 103)^2\cdot 235 \equiv -1 \Longrightarrow j_2 = 1$,

tehát $j = 101_2 = 5$ és $x \equiv b^j\cdot r \equiv 268^5\cdot 103 \equiv 304$ a keresett gyök. □

[KN]II.2.-ben (51.old.) megismerhetünk egy olyan algoritmust is, amely a fenti algoritmus és a Kínai Maradéktétel (és még néhány feltétel) segítségével összetett modulus esetén is tud négyzetgyököt vonni.

**6.94. Algoritmus.** ([KN],II.2.) **(vázlat)**

Legyen $m = p_1^{\alpha_1}p_2^{\alpha_2}\dots p_t^{\alpha_t}$ tetszőleges páratlan szám. *Amennyiben minden* $i \le t$ *esetén* ismerünk egy $n_i \in \mathbb{Z}_{p_i}$ kvadratikus nemmaradékot $(\mathrm{mod}\ p_i)$, akkor bármely $a \in \mathbb{Z}_m^*$, $m$-hez relatív prím négyzetes maradék esetén polinomidőben meg tudjuk oldani az (6.24) azaz

$$x^2 \equiv a \pmod{m}$$

kongruenciát az alábbiak szerint.

Először azt mutatjuk meg, hogy tetszőleges $p^\alpha$ prímhatvány modulusban hogyan lehet $a$-nak négyzetgyököt találni. Legyen tehát $x_0 \in \mathbb{Z}_p$ megoldása az

$$x^2 \equiv a \pmod{p}$$

kongruenciának. A megfelelő $(\mathrm{mod}\ p^\alpha)$ kongruencia megoldását keressük

$$x = x_0 + x_1p + \dots + x_{\alpha-1}p^{\alpha-1}$$

alakban.

Ha már mindegyik $i \le t$ esetén (külön-külön) megtaláltuk $a$ négyzetgyökét $(\mathrm{mod}\ p_i)$, akkor végül a Kínai Maradéktétellel megkapjuk $a$ négyzetgyökét $(\mathrm{mod}\ m)$. □

# 7. fejezet

# Kínai Maradéktétel és nagy számok szorzása

A fejezetben ismertetett eredményt 1000 évvel ezelőtt már valóban ismerték kínai matematikusok, és a modern algoritmikus számelméletben is „alapműveletként” használjuk.

## 7.1. Kínai Maradéktétel

**7.1. Probléma.** *Adott* $m_1, m_2, \ldots, m_r$, $a_1, a_2, \ldots, a_r \in \mathbb{Z}$ *egész számok esetén van-e az*

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \qquad \cdots \\ x \equiv a_r \pmod{m_r} \end{cases} \tag{7.1}$$

*ú.n.* ***szimultán*** *kongruenciarendszernek* $x \in \mathbb{Z}$ *gyöke?* $\square$

**7.2. Megjegyzés.** Az $r = 1$ esetet a 6.3. „Elsőfokú kongruencia-egyenletek” alfejezetben vizsgáltuk, tehát az alábbiakban $r \geq 2$.

Először a „legegyszerűbb” esettel foglalkozunk:

**7.3. Tétel** (Kínai Maradéktétel, KMT vagy CRT [Chinese Remainder/Residue Theorem])**.** *Ha*

$$\text{az} \quad m_i \quad \text{modulusok páronként relatív prímek,} \tag{7.2}$$

*akkor a (7.1) kongruenciarendszernek* bármilyen $a_1, a_2, \ldots, a_r \in \mathbb{Z}$ *egész számok esetén* pontosan egy $x$ *gyöke van* $\pmod{M}$ *ahol*

$$M = lkkt\,(m_1, m_2, \ldots, m_r)\,.$$

**Bizonyítás.** A bizonyítás képletet, sőt *gyors* **algoritmus**t is ad $x$ megtalálására.

Nyilván $M = m_1 \cdot m_2 \cdot \dots \cdot m_r$.

Először $x$ *egyértelműségét* mutatjuk meg. Ha $x_1$ és $x_2$ kielégítik a (7.1) kongruenciarendszert, akkor $x_1 \equiv x_2 \pmod{m_i}$ mindegyik $m_i$ modulusra, ahonnan a 6.14. Tétel szerint kapjuk, hogy $x_1 \equiv x_2 \pmod{M}$.

Most megadjuk $x$ *képletét*:

Először oldjuk meg (külön-külön) az

$$y_i \cdot \frac{M}{m_i} \equiv 1 \pmod{m_i}, \quad i = 1, 2, \dots, r$$

kongruenciákat (van megoldásuk, mert $\frac{M}{m_i}$ és $m_i$ relatív prímek mindegyik $i$-re a (7.2) feltétel miatt).

Ezután az

$$x := \sum_{i=1}^{r} a_i \cdot y_i \cdot \frac{M}{m_i} \tag{7.3}$$

képlet megadja a (7.1) kongruenciarendszer egy megoldását.

Algoritmus vége.

A fenti képlet helyességét az Olvasó könnyen beláthatja. ■

**7.4. Megjegyzés.** Az algoritmus gyors, polinomiális, hiszen az 5. „*Lineáris Diophantoszi egyenletek*” fejezetben megismertük a szükséges lineáris kongruenciák megoldását (Euklideszi algoritmussal).

**7.5. Példa.**
$$\begin{cases} x \equiv 5 \pmod 7 \\ x \equiv 2 \pmod{12} \\ x \equiv 3 \pmod{25} \\ x \equiv 0 \pmod{11} \end{cases}$$

**Megoldás:** A $7, 12, 25, 11$ modulusok páronként relatív prímek, ezért

$$M = lkkt(7, 12, 25, 11) = 7 \cdot 12 \cdot 25 \cdot 11 = 23\,100.$$

Az

$$y_i \cdot \frac{M}{m_i} \equiv 1 \pmod{m_i}$$

alakú kongruenciák

$$\begin{aligned} y_1 \cdot 3300 &\equiv 1 \pmod 7 \\ y_2 \cdot 1925 &\equiv 1 \pmod{12} \\ y_3 \cdot 924 &\equiv 1 \pmod{25} \\ y_4 \cdot 2100 &\equiv 1 \pmod{11} \end{aligned}$$

(egyik) megoldásai: $y_1 = -2 \equiv 5$, $y_2 = 5$, $y_3 = -1 \equiv 24$, $y_4 = -1 \equiv 10$.

Az (eredeti) kongruenciarendszer megoldása

$$\begin{aligned} x &\equiv \sum_{i=1}^{4} a_i y_i \frac{M}{m_i} \equiv 5 \cdot 5 \cdot 3300 + 2 \cdot 5 \cdot 1925 + 3 \cdot 24 \cdot 924 + 0 = \\ &\equiv 168\,278 \equiv 6578 \pmod{23\,100}. \quad \square \end{aligned}$$

Az [SzI1] feladatgyűjteményben sok kidolgozott példát találunk a Kínai Maradéktételre és alkalmazásaira. Például, a középiskolában jólismert "*Összefoglaló feladatgyűjtemény matematikából*" 3937. feladata így hangzik: "*Melyik az a legkisebb természetes szám, amely* 2 *-vel osztva* 1, 3 *-mal osztva* 2, 4 *-gyel osztva* 3 *és* 5 *-tel osztva* 4 *maradékot ad?* "

A jegyzethez mellékelt KINAI3D.EXE program segítségével gyakorolhatjuk az algoritmust.

Egyes programok (pl. Derive) beépített függvénye a CRT.

Mivel a maradékos osztás polinomoknál és komplex egészeknél is elvégezhető, így nem meglepő, hogy a Kínai Maradéktétel (pontosabban a fenti algoritmus) polinomoknál és komplex egészeknél is működik, kicsit persze fáradságosabb a polinomosztás miatt – amiben pedig a mellékelt POLIOSZ5.EXE program van segítségünkre.

## 7.2. Általános modulusok

Ha a modulusok nem (*páronként*) relatív prímek, akkor a feladat – a (7.1) kongruenciarendszer megoldása jóval nehezebb. ([SA1] 96-98. oldalán is csak a megoldás létezéséről történik említés.)

**Ötlet:** a (7.1) rendszerből kettő kongruenciát kiválasztunk, e kettő megoldását felírhatjuk egyetlen kongruenciában, tehát a kongruenciák számát lépésenként csökkenthetjük.

### r=2

**7.6. Tétel.** Tetszőleges $m_1, m_2 \in \mathbb{Z}$ *modulusok esetén az*

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \end{cases} \tag{7.4}$$

*kongruenciarendszer pontosan akkor oldható meg, ha*

$$lnko(m_1, m_2) \mid a_2 - a_1 \tag{7.5}$$

*és a megoldás egyértelmű* $\pmod{M_{1,2}}$ *ahol*

$$M_{1,2} := lkkt(m_1, m_2).$$

**Bizonyítás.** A bizonyítás ismét egy *gyors* **algoritmus**t is ad $x$ megtalálására.

A kongruenciák definíciója miatt (7.4) ekvivalens az

$$x = m_1 \cdot \ell_1 + a_1 = m_2 \cdot \ell_2 + a_2,$$

átrendezve az

$$m_1 \cdot \ell_1 - m_2 \cdot \ell_2 = a_2 - a_1 \tag{7.6}$$

lineáris Diophantikus egyenlettel. Innen látható, hogy a (7.4) kongruenciarendszer *megoldhatóságának* szükséges és elégséges feltétele valóban (7.5).

Legyen $d := lnko(m_1, m_2)$, ekkor a (7.6) egyenlet általános megoldása

$$\ell_1 = \ell_1^{(0)} + \frac{lkkt(m_1, m_2)}{m_1} \cdot t, \quad \ell_2 = \ell_2^{(0)} + \frac{lkkt(m_1, m_2)}{m_2} \cdot t \quad (t \in \mathbb{Z}) \,,$$

ahonnan (7.4) megoldása például:

$$x = m_1 \cdot \ell_1^{(0)} + lkkt(m_1, m_2) \cdot t + a_1 \qquad (t \in \mathbb{Z}) \,. \tag{7.7}$$

vagy ami ugyanaz:

$$x = m_2 \cdot \ell_2^{(0)} + lkkt(m_1, m_2) \cdot t + a_2 \quad (t \in \mathbb{Z}).$$

A megoldás valóban egyértelmű $(\mathrm{mod}\ M_{1,2})$. ■

**7.7. Példa.** Az

$$\begin{cases} x \equiv 3 & (\mathrm{mod}\ 6) \\ x \equiv 7 & (\mathrm{mod}\ 10) \end{cases}$$

egyenlet megoldása a következő:

A (7.6) egyenlet most $6\ell_1 - 10\ell_2 = 7 - 3 = 4$,

aminek megoldása $\ell_1 = -1 + 5 \cdot t, \quad \ell_2 = 1 + 3 \cdot t \quad (t \in \mathbb{Z})$.

Innen $x = -6 + 30t + 3 = 27 + 30t \quad (t \in \mathbb{Z})$, vagy másképpen

$x \equiv 27 \pmod{30}$. □

## r=3

Bár az alfejezet bevezetésében leírtuk, hogyan lehet a 7.6. Tétel segítségével a modulusok számát csökkenteni és így akárhány kongruenciából álló rendszert megoldani, az *r=3* eset egyszerű megoldóképletét mégis külön leírjuk.

A

$$\begin{cases} x \equiv a_1 & (\mathrm{mod}\ m_1) \\ x \equiv a_2 & (\mathrm{mod}\ m_2) \\ x \equiv a_3 & (\mathrm{mod}\ m_3) \end{cases} \tag{7.8}$$

kongruenciarendszer *általános* megoldása, *ha* a modulusok nem feltétlenül relatív prímek a következő:

A kongruenciák definíciója alapján a *kongruenciarendszer* ekvivalens az alábbi (lineáris Diophantikus) *egyenletrendszerrel*:

$$\begin{cases} x = m_1 \cdot \ell_1 + a_1 & (1) \\ x = m_2 \cdot \ell_2 + a_2 & (2) \\ x = m_3 \cdot \ell_3 + a_3 & (3) \end{cases}$$

valamilyen (megkeresendő) $\ell_1, \ell_2, \ell_3, x \in \mathbb{Z}$ egész számokra.

Az egyenleteket egymásból páronként kivonva kapjuk, hogy a megoldhatóság (egyik) *szükséges* feltétele:

$$lnko(m_i, m_j) \mid a_i - a_j \qquad (1 \leq i \neq j \leq 3). \tag{7.9}$$

(Ez nyilván tetszőleges számú kongruenciát tartalmazó rendszerre egy szükséges feltétel.)

Az előző feladat (7.7) végeredménye alapján az (1) és (2) egyenletek megoldása

$$x = m_1 \cdot \ell_1^{(0)} + L_{1,2.} \cdot t + a_1 \qquad (t \in \mathbb{Z})\,. \quad (4)$$

ahol

$$L_{1,2.} := lkkt(m_1, m_2)$$

és $\ell_1^{(0)}$ egyik gyöke az (1) és (2) egyenleteket tömörítő

$$m_1 \cdot \ell_1 - m_2 \cdot \ell_2 = a_2 - a_1 \tag{7.10}$$

egyenletnek (ld. (7.6) az előző feladatban).

Vagyis a (3) és (4) egyenletekből álló rendszert kell már csak megoldanunk (az ismeretlenek most: $x, \ell_3, t \in \mathbb{Z}$):

$$\begin{cases} x = m_3 \cdot \ell_3 + a_3 & (3) \\ x = L_{1,2.} \cdot t + \left(m_1 \cdot \ell_1^{(0)} + a_1\right) & (4) \end{cases}$$

A fenti egyenletrendszer megoldhatóságának *szükséges* feltétele (mint eddig):

$$lnko(m_3, L_{1,2.}) \mid a_3 - \left(m_1 \cdot \ell_1^{(0)} + a_1\right), \tag{7.11}$$

és a megoldás (ismét a (7.7) végeredmény alapján):

$$x = m_3 \cdot \ell_3^{(L)} + lkkt\left(L_{1,2}, m_3\right) \cdot s + a_3 \qquad (s \in \mathbb{Z}) \tag{7.12}$$

ahol $\ell_3^{(L)}$ egyik megoldása az

$$m_3 \cdot \ell_3 - L_{1,2.} \cdot t = a_3 - \left(m_1 \cdot \ell_1^{(0)} + a_1\right) \tag{7.13}$$

Diophantikus egyenletnek.

Érdemes még azt is észrevennünk, hogy

$$lkkt\left(L_{1,2}, m_3\right) = lkkt(m_1, m_2, m_3).$$

**7.8. Összegzés.** A (7.8) kongruenciarendszer megoldhatóságának szükséges és elégséges feltétele (7.9) és (7.11), gyökeit (7.12) adja meg, amelyhez előbb meg kell oldanunk a (7.10) és (7.13) egyenleteket.

**Az $r = 3$ eset vizsgálatának vége.** □

**7.9. Példa.** Az

$$\begin{cases} x \equiv 5 \pmod{6} \\ x \equiv 1 \pmod{10} \\ x \equiv 11 \pmod{15} \end{cases}$$

egyenlet megoldása a következő:

Először a (7.10) egyenletet kell megoldanunk:

$$6 \cdot \ell_1 - 10 \cdot \ell_2 = -4, \qquad \begin{cases} \ell_1 = -4 - 5u \\ \ell_2 = -2 - 3u \end{cases} \qquad (u \in \mathbb{Z}).$$

Ezután kapjuk a (7.13) egyenletet, megoldása:

$$15 \cdot \ell_3 - 30 \cdot t = 11 - (6 \cdot (-4) + 5) = 30, \qquad \begin{cases} \ell_3 = 2 - 2v \\ t \ \ = -v \end{cases} \qquad (v \in \mathbb{Z}).$$

Tehát a kongruenciarendszer megoldása:

$$x = 15 \cdot 2 + 30s + 11 = 41 + 30s \qquad (s \in \mathbb{Z})$$

vagyis

$$x \equiv 11 \pmod{30}. \quad \square$$

## 7.3. Nagy számok szorzása

Bár az 1.2. "*Alapműveletek sebesssége*" alfejezetben modern és gyors módszerekkel ismerkedtünk meg nagyméretű számok szorzására, hasznos lesz az alábbi, régi de nem elavult módszer is, a Kínai Maradéktétel felhasználásával.

**7.10. Algoritmus.** Adott $K$-nál kisebb természetes számok szorzása párhuzamos (szimultán) módszerrel.

Rögzítsünk páronként relatív prím $m_1, m_2, \ldots, m_r \in \mathbb{N}$ számokat úgy, hogy a várható végeredmény $M = lkkt(m_1, m_2, \ldots, m_r) = m_1 \cdot m_2 \cdot \cdots \cdot m_r$ alatt maradjon, vagyis $K^2 < M$ legyen (hiszen $X, Z < K$ esetén $X \cdot Z < K^2$).
Még a műveletek megkezdése **előtt** kiszámíthatjuk és eltároljuk a

$$y_i \cdot \frac{M}{m_i} \equiv 1 \pmod{m_i}, \quad i = 1, 2, \ldots, r \tag{7.14}$$

kongruenciák megoldásait, vagyis az $y_i$ számokat és az $y_i \cdot \dfrac{M}{m_i}$ konstansokat ($i = 1, 2, \ldots, r$).

Ha most kapunk összeszorzandó $X, Z < K$ számokat, akkor már egyszerre használhatunk $r$ számítógépet: először kiszámítjuk az

$$x_i \equiv X \text{ és } z_i \equiv Z \pmod{m_i} \qquad i = 1, 2, \ldots, r$$

és az

$$a_i :\equiv x_i \cdot z_i \pmod{m_i} \qquad i = 1, 2, \ldots, r$$

értékeket. Ekkor az

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \qquad \cdots \\ x \equiv a_r \pmod{m_r} \end{cases}$$

kongruenciarendszert a 7.3. Tételben ismertetett algoritmussal megoldhatjuk, és a megoldás nyilván

$$x \equiv X \cdot Z \pmod{M}$$

vagyis

$$x = X \cdot Z.$$

A 7.10. Algoritmus vége.
□

**7.11. Megjegyzés.** Vegyük észre, hogy az $y_i$ számokat csak egyszer kell kiszámolnunk, továbbá az $x_i$, $z_i$ és $a_i$ számokat egyszerre több számítógépen (párhuzamosan) is kiszámíthatjuk. (Ezért az $m_i$ számokat nyilván optimálisan kell kiválasztanunk: sem túl kicsi sem túl nagy $m_i$ modulusok nem jók.) Végül $x$ kiszámítása a (7.3) képlet szerint mindössze csak pár konstans szorzása kis számokkal és összeadás.

**7.12. Példa.** Az

$$\begin{cases} m_1 = 253\ , \\ m_2 = 200\ , \\ m_3 = 261\ , \\ m_4 = 247\ . \end{cases}$$

modulus rendszer segítségével számítsuk ki a következő „nagyméretű” szorzásokat:

**a)** $X_1 = 56\,079, \quad Z_1 = 58\,144,$

**b)** $X_2 = 49\,745, \quad Z_2 = 55\,846,$

**c)** $X_3 = 57\,898, \quad Z_3 = 48\,653,$

**d)** $X_3 = 56\,898, \quad Z_3 = 49\,866.$

**Megoldás:** Mindenütt igyekszünk a legkisebb abszolút értékekkel (azaz negatív maradékokkal is) számolni.

**(0) Előkészítés:** Az Euklideszi algoritmussal könnyen ellenőrizhető, hogy a megadott $m_i$ modulusok páronként relatív prímek.

$$M = 253 \cdot 200 \cdot 261 \cdot 247 = 3\,262\,030\,200.$$

A (7.14) kongruenciák megoldásai

$$y_1 = -18 \implies y_1 \cdot \frac{M}{m_1} = -232\,081\,200,$$

$$y_2 = -49 \implies y_2 \cdot \frac{M}{m_2} = -799\,197\,399,$$

$$y_3 = 17 \implies y_3 \cdot \frac{M}{m_3} = 212\,469\,400,$$

$$y_4 = 62 \implies y_4 \cdot \frac{M}{m_4} = 818\,809\,200.$$

**(1) A tényleges szorzások:**

**a)** input: $X = 56079, \quad Z = 58144,$

**(i)** a párhuzamos számolások:

$$\begin{cases} x_1 := X \equiv 166 \pmod{m_1} \\ x_2 := X \equiv 79 \pmod{m_2} \\ x_3 := X \equiv -44 \pmod{m_3} \\ x_4 := X \equiv 10 \pmod{m_4} \end{cases}, \quad \begin{cases} z_1 := Z \equiv -46 \pmod{m_1} \\ z_2 := Z \equiv -56 \pmod{m_2} \\ z_3 := Z \equiv -59 \pmod{m_3} \\ z_4 := Z \equiv 99 \pmod{m_4} \end{cases}$$

majd

$$\begin{cases} x_1 \cdot z_1 \equiv -46 \pmod{m_1} \\ x_2 \cdot z_2 \equiv -24 \pmod{m_2} \\ x_3 \cdot z_3 \equiv 36 \pmod{m_3} \\ x_4 \cdot z_4 \equiv 2 \pmod{m_4} \end{cases}$$

**(ii)** az összesítés:

$$\begin{aligned} X \cdot Z &= \sum_{i=1}^{t} y_i \cdot \frac{M}{m_i} \cdot x_i \cdot z_i = \\ &= (-232\,081\,200) \cdot (-46) + (-799\,197\,399) \cdot (-24) + 212\,469\,400 \cdot 36 \\ &\quad + 818\,809\,200 \cdot 2 \\ &= 39\,142\,989\,576 \equiv 3\,260\,657\,376 \pmod{M} \end{aligned}$$

Az eredmény jó, mert valóban X·Z = 3 260 657 376.

□

# 8. fejezet

# Prímtesztelés és számok felbontása

Mint a 3.2. „*A számelmélet algoritmikus problémái*” alfejezet 3.16. Problémájában láttuk, a *prímfelbontás* (*faktorizáció*) és a *prímtesztelés* problémák messze nem azonosak (különösen az AKS algoritmus [ld. 8.6. alfejezet] felfedezése óta). Azonban a klasszikus algoritmusok hasonló módszereket használnak a két problémára, ezért tárgyaljuk e két problémát egy fejezetben.

Látni fogjuk, hogy a *prímfelbontó* módszerek mindegyike $\mathcal{O}\left(2^n\right)$ *exponenciálisan lassú* (a matematikusok az $n \to \infty$ „határesetet” vizsgálják), csak „kicsi” (100-200 jegyű) számokra használhatóak a gyakorlatban. A konstansok különbözősége miatt azonban előfordulhat, hogy egyik módszer évmilliárdokig, míg a másik „csak” évmilliókig fut ugyanazon adat esetén. Szemléletes adatokat találunk a 8.2. Példában, 10.23. Megoldásban és [JA]-ban.

Agrawal, Kayal és Saxena 2001-ben feltalált algoritmusa áttörést hozott: polinomiálisan gyors, 100% biztonságos, determinisztikus algoritmus a *prímtesztelés* problémára (a gyors *prímfelbontás* problémája máig is megoldatlan). Az utolsó alfejezetben csak röviden ismertetjük az „AKS” algoritmust, mert több matematikai előismeretet követel, mint amennyit jelen könyvünk tartalmaz. „Kis” számokra a régebbi (könyvünkben ismertetett) algoritmusok jól használhatók.

## 8.1. Eratoszthenesz algoritmusa

Többezer éves, iskolában tanult algoritmus:

**8.1. Algoritmus. Eratoszthenész algoritmusa**:
Legyen az input egy *tetszőleges* (többszázjegyű) $n \in \mathbb{N}$ természetes szám. Osszuk el $n$-et 2-vel és az $\sqrt{n}$-nél kisebb páratlan számokkal. Ha valamelyik osztás nem ad maradékot, akkor $n$ nem prím és meg is kaptuk $n$ egy felbontását. Ellenkező esetben $n \in \mathbb{P}$ prímszám. □

**8.2. Megjegyzés.** Számoljunk utána: mennyi lépés (mennyi mp) egy-egy szám felbontása a mai számítógéppekkel: ezt részletesen elemeztük a 2.2. „A számelmélet algoritmikus problémái” alfejezet 3.21. Példájában. □

Tehát az Eratoszthenészi algoritmus nagyon lassú (exponenciális). Mint a 3.21. Példában láttuk: elemi módszerekkel sem gyorsíthatunk rajta lényegesen. [MGy] szerint a modern

programok osztások helyett kivonásokat ismételgetnek, de többszázjegyű számoknál ez még mindig exponenciálisan lassú.

## 8.2. Fermat algoritmusa

*Pierre Fermat* (1601-1664) algoritmusa ma is hatékony párszáz-jegyű számokra, különösen akkor, ha a felbontandó szám két közeli prím szorzata. Az algoritmus összetett számoknál (sikeres futás esetén) egy felbontást is megad. (Tehát titkosírásnál sem célszerű olyan számot választanunk, ami két közeli prím szorzata.)

Tehát adott egy tetszőleges (nagy) $n \in \mathbb{Z}$ felbontandó szám.

Fermat *első ötlete*: *ha* $n$-nek van két közeli osztója: $n = ab$ és $y = a - b$ kicsi ($a > b$), akkor az $x = \dfrac{a+b}{2}$ jelöléssel $a = x + y$ és $b = x - y$, és ekkor alkalmazhatjuk az $n = ab = (x+y)(x-y) = x^2 - y^2$ azonosságot.

Tehát keressük $n$-et

$$n = ab = (x+y)(x-y) = x^2 - y^2 \tag{8.1}$$

alakban.

Nyilvánvalóan az $x \approx \sqrt{n}$ vagyis az $x \approx [\sqrt{n}]$ értékből kiindulva kezdjük a számításokat, $x$-et egyesével növelve, továbbá $0 \leq y < x < \dfrac{n}{2}$.

Fermat azt is észrevette (*második ötlet*), hogy a (8.1) egyenlőséget

$$x^2 - n = y^2$$

alakban írjuk, akkor első közelítésként elegendő $x^2$ *utolsó két számjegyét* tekinteni (vagyis csak $x$ *utolsó két számjegyét*) mivel $y^2$ utolsó két számjegye csak $00$, $e1$, $e4$, $25$, $o6$ és $e9$ lehet ($e$ páros, $o$ páratlan számjegyet jelöl), és $n$ rögzített. Vagyis (még négyzetre emelés és gyökvonás előtt) a lehetséges $\sqrt{n} < x < \dfrac{n}{2}$ számoknak legalább a $78/100$-része kiesik.

Az „utolsó *két* számjegy” $(\mathrm{mod}\ 100)$ vizsgálatot jelent, több modulussal még tovább szűkíthetjük a lehetséges $x$ számok körét: legyenek $m_1, \ldots m_K$ rögzített tetszőleges modulusok, keressük meg mindegyik $i \leq K$ esetén a négyzetszámok $(\mathrm{mod}\ m_i)$ maradékait – esetleg táblázatban is tárolhatjuk őket, amely halmazok meghatározzák $x$ lehetséges maradékait $(\mathrm{mod}\ m_i)$ *minden* $i \leq K$ esetén. Így már $x$ négyzetreemelése előtt a biztosan rosszakat eleve kiszűrhetjük, szinte alig marad $x$ és $y$ a (8.1) egyenlőség kipróbálására (ahol persze $y$ meghatározásához szükségünk van egy négyzetgyökvonásra is).

**8.3. Algoritmus. (Fermat algoritmusa)**
*Előkészítés:* Legyenek $m_1, \ldots, m_K$ páronként relatív prím és $n$-hez is relatív prím modulusok. Készítsünk el $K$ darab (az adott $n$-től függő) $S[i,j]$ szita táblázatot:
legyen $1 \leq i \leq K$, $0 \leq j < m_i$ esetén
$S[i,j] := 1$ ha *van olyan* $y$ amelyre $j^2 - n \equiv y^2 \pmod{m_i}$, és
$S[i,j] := 0$ más esetekben.
*Az algoritmus:* Legyen $x$ kezdeti értéke $x := [\sqrt{n}]$.

$x$-et egyesével növelve először ellenőrizzük:
ha $S[i,\, x \pmod{m_i}] = 0$ valamely $1 \le i \le K$ esetén, akkor növeljük $x$-et,
ha $S[i,\, x \pmod{m_i}] = 1$ minden $1 \le i \le K$ esetén, akkor ellenőrizzük, hogy $x^2 - n$ négyzetszám-e. □

**8.4. Megjegyzés.** Ez az algoritmus még mindig exponenciális, de *Eratoszthenész* algoritmusánál *nagyságrendekkel* jobb (pedig csak a konstans szorzót javítottuk). Az eljárás bitműveletekkel gyorsítható, de még így sem elég gyors. Alapgondolata sok mai eljárásban felbukkan. A módszerrel már 1965-ben egymillió próba/másodperc sebességet értek el: elektromechanikus (fogaskerék, biciklilánc) gépeket szerkesztettek szitálásra ([JA]).

Bár a felbontandó $n$ eleve csak páratlan szám, az alábbi Tételt nem árt tudnunk:

**8.5. Tétel.** *Az* $n = x^2 - y^2$ *egyenletnek akkor és csak akkor léteznek* $x, y \in \mathbb{Z}$ *gyökei, ha* $n \neq 4k + 2$.

**Bizonyítás.** $\boxed{\Rightarrow}$ $n = x^2 - y^2 = (x + y)(x - y)$ esetén tudjuk, hogy $(x + y)$ és $(x - y)$ párossága (paritása ) ugyanaz (vagy mindkettő páros vagy mindkettő páratlan), ezért $n$ nem lehet $4k + 2$ alakú.
$\boxed{\Leftarrow}$ $n \neq 4k + 2$ esetén találunk olyan $n = n_1 \cdot n_2$ felbontást, amelyre $n_1$ és $n_2$ párossága ugyanaz. Ekkor az

$$\begin{cases} x - y = n_1 \\ x + y = n_2 \end{cases}$$

egyenletrendszernek van megoldása, hiszen $x = \dfrac{n_1 + n_2}{2}$, $y = \dfrac{n_1 - n_2}{2}$. ■

A módszer tovább gyorsítható az ún. *faktorbázisok* ötletével (*harmadik ötlet*), melyet most csak röviden vázolunk.

**8.6. Algoritmus. Legendre–Kraitchik módszer** (vázlat)
Az $x^2 - y^2 = n$ (8.1) egyenlet helyett, ha találnánk olyan $x, y \in \mathbb{Z}$, $x \neq \pm y$ számokat, amelyekre

$$x^2 - y^2 = 0 \pmod{n}, \tag{8.2}$$

akkor $lnko(n,\, x + y)$ vagy $lnko(n,\, x - y)$ adná $n$-nek egy (valódi) osztóját.
Ilyen $x$ és $y$ meglehetősen gyors keresésére válasszunk aránylag kis prímek egy kicsi $B = \{p_1, p_2, \ldots, p_h\}$ halmazát – ezt nevezzük **faktorbázis**nak. Ha $x$ és $y$-t $B$ elemeinek szorzataként keressük, akkor a $p_i$ prímek kitevőiből egy lineáris egyenletrendszert írhatunk fel, melynek gyors megoldásából tudunk (8.2) megoldásaira következtetni. (További részletes példák [KN] 132-143.oldalain találhatóak.) □

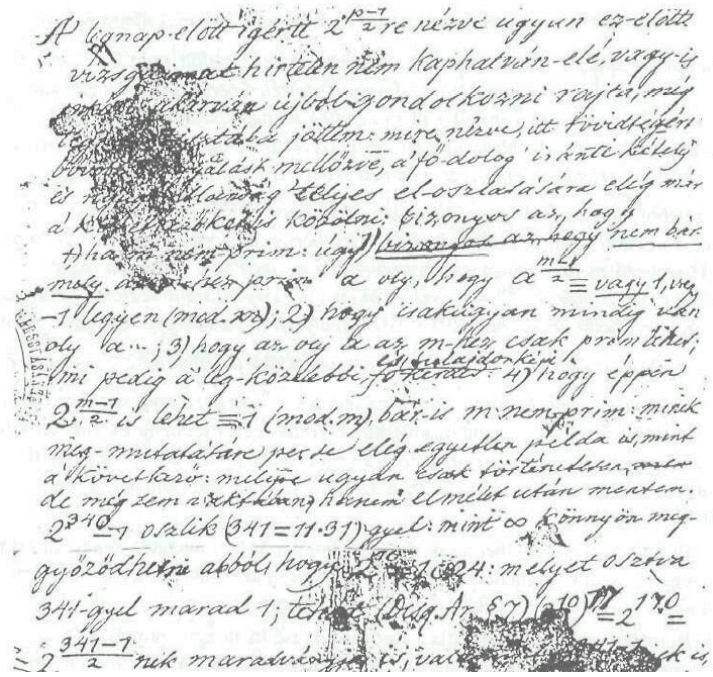
## 8.3. Álprímek

Valamennyi Bolyai-monográfia szerzőjének véleménye szerint Bolyai János a számelmélet terén nem ért el semmilyen említésre méltó eredményt. Kéziratos hagyatékának lapjai ennek a véleménynek éppen az ellenkezőjéről tanúskodnak: Bolyai Jánost a számelmélet valósággal elbűvölte. Különösen a prímszámokkal kapcsolatos kérdések kötötték le a figyelmét: egy

olyan eljárást keresett, amelynek segítségével bármely racionális prímszám megfelelő *képlettel* (”**prímképlet**”) kifejezhető (a $\mathbb{Z}[i]$-beli prímeket Bolyai megtalálta). Apja, Bolyai Farkas ösztönzésére megpróbálta bebizonyítani a 6.54. „kis” Fermat Tétel fordítottját – ld. az alábbi 8.7. Problémát. Bolyai azonban néhány kísérlet után *több* olyan *összetett* számra bukkant, amelyekre (8.3) igaz. Azt találta például, hogy

$$2^{340} \equiv 1 \pmod{341} \quad \text{és} \quad 4^{14} \equiv 1 \pmod{15}$$

pedig $341 = 11 \cdot 31$ és $15 = 3 \cdot 5$.



Bolyai János levele [KE2] (A Typotex Kiadó engedélyével)

A kis Fermat tételt kielégítő *összetett* számokat *pszeudoprímszámoknak*, *álprímeknek* nevezzük. Bolyai János tehát felfedezett több álprímet. A fenti vizsgálódásai során fedezte fel Bolyai János a már említett 6.55. Tételt is. (Kiss Elemér [KE1])

**8.7. Probléma.** *Igaz-e, hogy* ***ha*** *egy* $n \in \mathbb{Z}$ *szám teljesíti a következő feltételt:*

$$b^{n-1} \equiv 1 \pmod{n}, \quad \textbf{\textit{minden}}\ 1 < b < n, \quad n\text{-hez relatív prím számra} \tag{8.3}$$

***akkor*** $n$ *prímszám?* □

Az alább következő algoritmusnak a szakirodalomban semmilyen elnevezését nem találtuk, ezért neveztük el mi *Bolyai-teszt*nek, mert a „Bolyai-algoritmus” elnevezés már foglalt[1)].

[1)] **Bolyai Farkas tétele**: Az $x^m = x + a$ $(m > 2)$ ún. „trinom” egyenletek közelítő (rekurzív) megoldására: legyen $x_1 := \sqrt[m]{a}$ és $x_{n+1} := \sqrt[m]{a + x_n}$ $(n = 1, 2, \dots)$. Ekkor $\lim\limits_{n\to\infty} x_n = x^*$ ahol $x^*$ az $x^m = x + a$ egyenlet egyik gyöke. □

**8.8. Algoritmus. Bolyai-teszt:** *(Ötlet)* Tetszőleges $n$ szám esetén keressünk olyan $1 < b < n$, $n$-hez relatív prím számot, amelyre

$$b^{n-1} \not\equiv 1 \pmod{n}. \tag{8.4}$$

Ha találunk ilyen $b$ számot, akkor $n$ nyilván nem prímszám, vagyis összetett.
Ha pedig minden ilyen $b$ számra (8.4) *nem* teljesül, akkor $n$ *talán* prímszám? □

**8.9. Definíció.** Ha $b < n$, $b$ relatív prím $n$-hez olyan szám, amelyre (8.4) teljesül, akkor $b$-t $n$ **árulójá**nak nevezzük. □

Hangsúlyozzuk, hogy bár a fenti Algoritmus a kis-Fermat tételen alapul, mégsem ez Fermat Algoritmusa – amit az előző alfejezetben ismertettünk!

Az Olvasó bizonyára észrevette: az eljárás nem adja meg az $n$ szám egyetlen osztóját sem, csak "$n$ *biztosan összetett*" vagy "$n$ *valószínűleg prím*" válaszok valamelyikét – tehát csak prímtesztelő eljárás.

**8.10. Megjegyzés.** Mint minden algoritmusnál, egyik fontos tényező a futásidő. Az $lnko\,(n, b)$ és $b^{n-1} \pmod{n}$ mennyiségeket gyorsan ki tudjuk számítani, a 4.2. „Euklidesz algoritmusa" és a 6.6. „Nagy kitevőjű hatványozás" alfejezetek alapján.
Azonban nem tudjuk az *összes,* $n$-nél kisebb $b$-t megvizsgálni. Ha egy $b$ árulót találunk, akkor persze már OK. Hány áruló van 1-től $n$-ig? Erre a kérdésre még visszatérünk.

Még meglepőbb: vannak olyan *összetett* számok, melyeknek egyetlen árulójuk sincs: *minden* $b < n$, $n$-hez relatív prím számra $b^{n-1} \equiv 1 \pmod{n}$, azaz (8.3) teljesül!

**8.11. Definíció. (i)** Az $n \in \mathbb{Z}$ páratlan *összetett* számot **álprím**nek (**pszeudoprím**nek) nevezzük a $b$ "**bázis**" alapján, ha

$$lnko\,(b, n) = 1 \quad \text{és} \quad b^{n-1} \equiv 1 \pmod{n}$$

teljesül. Használatos még a „$b$ **cinkosa** $n$-nek" elnevezés is.

**(ii)** Az $n \in \mathbb{Z}$ *összetett* számot **Carmichael-számnak** nevezzük, ha $n$ **álprím** *minden* $b < n$, $n$-hez relatív prím számra, azaz teljesül a (8.3) feltétel. □

Csak a múlt (XX.) században derült ki, hogy *végtelen sok* álprím létezik, például $n = 15$, 91, 341, rendre a $b = 4$, 2, 3 bázisokkal.

Ráadásul *Robert Daniel Carmichael (1879-1967) amerikai matematikus* 1910-ben már legalább 15 olyan számot talált, amelyre a 8.11.(ii) Definíció teljesül, azaz Carmichael-szám. Például $n = 561$, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, .... Az összes $b$-re: végigpróbálása helyett ajánljuk inkább Korselt 8.15 Tételét alább.

Évtizedekig megoldatlan volt, hogy létezik-e *végtelen sok* Carmichael (=„rossz") szám. Végül, *Erdős Pál (1913-1996)* egy ötletének felhasználásával *Alford, Granville* és *Pomerance* 1994-ben bizonyította be, hogy végtelen sok Carmichael szám van.

Tehát a fenti 8.8. Algoritmus nem 100% módszer! Nézzük a további részleteket:

**8.12. Tétel.**
***(i)*** *$n$ álprím a $b$ bázisra akkor és csak akkor ha $b$ rendje $\mathbb{Z}_n^*$-ban osztója $n-1$-nek.*
***(ii)*** *Ha $n$ álprím a $b_1$ és $b_2$ bázisokra, akkor a $b_1b_2$ és $b_1b_2^{-1}$ bázisokra is álprím (ahol $b^{-1}$ a $b \in \mathbb{Z}_n^*$ multiplikatív inverze).*
***(iii)*** *Ha $n$ nem álprím (legalább egy $b$ bázisra), akkor a lehetséges $b \in \mathbb{Z}_n^*$ számok legalább a felére sem álprím.*

**Bizonyítás. (i)** és **(ii)** nyilvánvaló.
**(iii)** Legyen $n$ álprím a $b_1, \ldots, b_s \in \mathbb{Z}_n^*$ számokra, és legyen $b \in \mathbb{Z}_n^*$ olyan szám, amelyre $n$ *nem* álprím. Ha $n$ valamelyik $bb_i$ szorzatra álprím lenne, akkor (ii) szerint a $(bb_i)b_i^{-1} = b$-re is, ami a feltevésnek ellentmond. Tehát $n$ *nem* álprím az összes $bb_i$ szorzatra. ■

Most már rátérhetünk több más fontos kérdésre: hogyan válasszuk a $b$-t, mennyi $b$-t válasszunk, mekkora a módszer megbízhatósága?

**8.13. Algoritmus.** *(Éles)*
véletlenszerűen választunk egy (tetszőleges) $b$ számot 2 és $n-1$ között,
kiszámítjuk $d := lnko\,(n, b)$-t az Euklideszi algoritmussal,
ha $d > 1$ akkor nyilván $n$ összetett (sőt $d$ egy nemtriviális osztója $n$-nek), STOP;
ha $d = 1$ (vagyis $b \in \mathbb{Z}_n^*$) akkor kiszámítjuk $e :\equiv b^{n-1}$ értékét $(\mathrm{mod}\ n)$,
ha $e > 1$ akkor nyilván $n$ összetett, STOP;
ha $e = 1$ akkor próbálkozzunk egy másik $b$ számmal: kezdjük előlről. □

**8.14. Megjegyzés.** Ha történetesen $n$ *nem* prímszám és *nem* Carmichael szám, akkor $k$ db $b$ „sikertelen" szám kipróbálása után, a 8.12. Tétel (iii) alapján biztosan mondhatjuk: "$n$ legfeljebb $2^{-k}$ eséllyel prím". Sajnos vannak Carmichael számok is a világon, tehát módszerünk soha nem mondhatja 100% biztonsággal, hogy "$n$ prím".
*Ha véletlen módszerrel választunk egy legfeljebb 13 jegyű pozitív egészt, akkor 3,46 % az esélye, hogy prím lesz – ez egyáltalán nem elhanyagolható –, és ha megfelel a 2-es alapú teszten, akkor kb. 99,9999236% ([FR]) a valószínűsége, hogy tényleg prím [FR]. Ez azt mutatja, hogy a véletlen segítségével gyorsan találhatunk olyan nagy számot, ami igen nagy valószínűséggel prím. A módszer még erősíthető is, a biztonság tetszőlegesen növelhető.*

**8.15. Tétel.** *Tetszőleges $n$ páratlan számra*
***(i)*** *Ha $n$ négyzetszámmal osztható, akkor nem Carmichael szám.*
***(ii)*** ***(Korselt**, 1899): Egy $n$ négyzetmentes szám akkor és csak akkor Carmichael szám, ha*

$$p - 1 \mid n - 1$$

*minden $p \mid n$ prímszámra.*
***(iii)*** *Minden Carmichael szám legalább három (különböző) prímszám szorzata.*

**Bizonyítás. (i)** lásd [KN] 115.oldalán.
**(ii)** egyik fele: Ha $n$ minden $p$ prímosztójára $p-1 \mid n-1$ akkor minden $b \in \mathbb{Z}_n^*$ számra $b^{n-1} \equiv (b^{p-1})^s \equiv 1\ (\mathrm{mod}\ p)$, vagyis $b^{n-1} - 1$ osztható $n$ minden $p$ prímosztójával, vagyis $n$-nel is. Tehát $n$ valóban Carmichael szám. A megfordítás bizonyítását ld. [KN] 115.oldalán.

**(iii)** Ha $n = pq$, $p < q$ és $n$ Carmichael szám lenne, akkor (ii) alapján $q-1 \mid n-1 = p(q-1)+(p-1)$ azaz $q-1 \mid p-1$ lenne, ami ellentmondás. ■

Tehát $n = 561 = 3 \times 11 \times 17$ valóban Carmichael szám, hiszen $3-1 \mid 560$, $11-1 \mid 560$ és $17-1 \mid 560$.

Persze a fenti Tételt nem tudjuk alkalmazni akkor, ha $n$ prímtényezős felbontását nem ismerjük, de legalább Carmichael számokat tudunk keresni ....

Az érdeklődő Olvasóknak még **Kiss Elemér** [KE1], **Freud Róbert** [FR] és **Járási István** [JI] cikkeit valamint **Koblitz,N.** [KN] könyvét ajánljuk.

## 8.4. Miller–Rabin teszt

A *Miller–Rabin teszt* a 7.3. "*Álprímek*" alfejezetben megismert 8.8. Bolyai-teszt továbbfejlesztett változata: még biztosabb % eredményt ad. Szintén nem adja meg az $n$ szám egyetlen osztóját sem, csak "$n$ *biztosan összetett*" vagy "$n$ *valószínűleg prím*" válaszok valamelyikét – tehát megint egy prímtesztelő eljárást ismerhetünk meg.

**Ötlet:** ha $n$ prímszám (lenne), akkor 1-nek csak $+1$ és $-1$ a négyzetgyökei, azaz az $x^2 \equiv 1 \pmod{n}$ egyenletnek csak $x \equiv \pm 1 \pmod{n}$ a megoldásai (az 5.8. "*Magasabbfokú kongruenciák*" alfejezet 6.79. Állítása szerint, másképpen: $\mathbb{Z}_p$ test).

**8.16. Algoritmus.** (csak tervezgetés): $n \in \mathbb{N}$ adott felbontandó szám, $b < n$, $lnko(b,n) = 1$ és

$$b^{n-1} \equiv 1 \pmod{n}$$

($n$ *álprím* a $b$ bázisra).
Mivel $n-1$ páratlan, ezért $b^{\frac{n-1}{2}} \equiv \pm 1 \pmod{n}$ kell lennie.
Ha $b^{\frac{n-1}{2}} \not\equiv \pm 1 \pmod{n}$ akkor $n$ biztosan összetett, STOP.
Ha $b^{\frac{n-1}{2}} \equiv -1 \pmod{n}$, akkor $b$-vel már nem tudunk továbbmenni, választunk egy következő $b$-t.
Ha $b^{\frac{n-1}{2}} \equiv +1 \pmod{n}$ akkor próbálkozzunk a $b^{\frac{n-1}{4}} \stackrel{?}{\equiv} \pm 1 \pmod{n}$ kérdéssel,
... és így tovább ... egészen addig, míg a kitevő $\dfrac{n-1}{2^s}$ páratlan nem lesz. □

A gyakorlatban ezt „visszafelé" érdemes számolni, hiszen nagy kitevőjű hatványokat a kitevő *növelésével* szoktuk kiszámítani.

**8.17. Algoritmus. (Miller–Rabin teszt)**
Legyen $n \in \mathbb{N}$ adott felbontandó szám.
Jelölje $s$ a legnagyobb kitevőjét 2-nek $n-1$-ben:

$$2^s \parallel n-1$$

amikor

$$t := \frac{n-1}{2^s}$$

páratlan egész szám.
Válasszunk egy $b < n$ számot melyre $lnko\,(b, n) = 1$.
Egymás után, sorban számítsuk ki a $b^t$, $b^{2t}$, $b^{4t}$, $\ldots$, $b^{2^s t} = b^{n-1}$ hatványokat $(\mathrm{mod}\ n)$.
Amikor legelőször 1-et kapunk, megnézzük: előtte $-1$ kell lennie, különben $n$ biztosan összetett. Kicsit pontosabban: ha $0 < r \leq n-1$ a *legkisebb* olyan kitevő, amelyre

$$b^{2^r \cdot t} \equiv +1 \pmod{n}$$

akkor ellenőrizzük a

$$b^{2^{r-1} \cdot t} \stackrel{?}{\equiv} -1 \pmod{n}$$

feltételt.
Ha $b^{2^{r-1} \cdot t} \not\equiv -1 \pmod{n}$ akkor $n$ biztosan összetett, STOP.

Ha $b^{2^{r-1} \cdot t} \equiv -1 \pmod{n}$ akkor $n$ összetett/prím tulajdonságáról semmit sem tudunk: próbálkozzunk másik $b$ számmal. □

**8.18. Definíció.** Ha $n$, $b$, $s$ és $t$ a fenti 8.17. Algoritmusban leírtak, és:
vagy $b^t \equiv 1 \pmod{n}$ vagy van $0 \leq r < s$ amelyre $b^{2^r \cdot t} \equiv -1 \pmod{n}$,
akkor $n$-et **erős álprím**nek nevezzük a $b$ **bázisra** vonatkozóan ($b$ **erős cinkosa** $n$-nek, stb.). □

**8.19. Példa.** Megmutatjuk, hogy $n = 91 = 7 \times 13$ erős álprím a $b = 10$ bázisra vonatkozóan. Ekkor $n - 1 = 90 = 2^1 \cdot 45$, $s = 1$, $t = 45$. Mivel $10^3 = 1001 - 1 \equiv -1 \pmod{91}$, ezért $b^t = 10^{45} \equiv (-1)^{15} \equiv -1 \pmod{91}$ alapján $r = 0$ bizonyítja állításunkat. □

Az algoritmusról további részleteket [KN] 116-122 oldalain olvashatunk.

## 8.5. Pollard $\rho$-módszere

Pollard módszere egy valódi **osztót** is keres, de az Eratoszstheneszi osztogatásnál lényegesen gyorsabb. Szokás "*Monte Carlo*" módszernek is hívni, Pollard 1975-ös cikkének eredeti címe miatt. (Nem tévesztendő össze Pollard „$p-1$"-módszerével, amit könyvünkben nem tárgyalunk.)

**8.20. Algoritmus.** Legyen $f\,(x)$ egy polinom. Tulajdonképpen számunkra
$f\colon \mathbb{Z}_n \to \mathbb{Z}_n$ csak egy leképezés, vagyis a vakpróbálkozást irányítja, később tárgyaljuk milyen $f\,(x)$ az optimális. Pl. $f\,(x) = x^2 + 1$ jó szokott lenni.
Legyen $x_0 < n$ tetszőleges, legyen $1 < k$ esetén $x_k :\equiv f(x_{k-1}) \pmod{n}$, és vizsgáljuk az $x_k$-kat: különböző maradékosztályokba esnek-e $(\mathrm{mod}\ n)$, ezenkívül $lnko\,(x_k - x_j,\, n) > 1$-et vizsgáljuk meg *minden (vagy esetleg néhány)* $j < k$-ra.
Nyilván $r = lnko\,(x_k - x_j,\, n) > 1$ esetén $r$ egy valódi osztója $n$-nek.

**8.21. Megjegyzés. (o)** Nyilvánvaló, hogy $x_k$ és $x_j$ nem eshetnek egyazon maradékosztályba hiszen ekkor $x_k - x_j \equiv 0 \pmod{n}$. Persze, $(\mathrm{mod}\ r)$ ugyanabba az osztályba kellene esniük, mert ekkor $r$ egy valódi osztója $n$-nek, vagyis $lnko\,(x_k - x_j,\, n) = r > 1$.

**(i)** Ha $k$ már nagy és növekszik, akkor túl sok a kipróbálandó $j$-k száma: $\binom{k}{2} = \mathcal{O}(k^2)$, ezt az alábbiakban ki tudjuk kerülni.
**(ii)** Azt is meg fogjuk vizsgálni, hogy milyen $f$ függvényekkel lehet a módszert gyorsítani.

**8.22. Segédállítás.** *Legyen $S$ egy $r$-elemű halmaz és $\ell < r$ rögzített szám. Tekintsük az olyan $g : S \to S$ függvényeket és $s_0 \in S$ elemeket, amelyekre az $s_0, s_1, \ldots, s_\ell \in S$ elemek mind különbözőek, ahol $s_j = g(s_{j-1})$ ha $j = 1, 2, \ldots$. Ekkor az ilyen $(g, s_0)$ párok aránya az összes lehetséges $(g, s_0)$ párhoz viszonyítva kisebb, mint $e^{-\lambda}$, ahol $\lambda = \dfrac{(\ell-1)^2}{2r}$.*

**Bizonyítás.** Elemi kombinatorika segítségével. ■

**8.23. Állítás.** *Ha a bevezetőben említett $x_j$ sorozatban $x_{k_0} \equiv x_{j_0} \pmod r$ valamely $j_0$, $k_0$ párra és $r$ számra, akkor a továbbiakban minden $t \in \mathbb{N}$ esetén szintén $x_{k_0+t} \equiv x_{j_0+t} \pmod r$. Más szavakkal: ha $k > k_0$ és $k - j = k_0 - j_0$ akkor $x_k \equiv x_j \pmod r$.* □

**8.24. Algoritmus** (Gyorsított változat)**.** A sorozat minden

$$x_k \equiv f(x_{k-1}) \pmod n$$

tagjának kiszámítása után nem kell az összes előző $j < k$ indexre ellenőrizni az $lnko(x_k - x_j, n) > 1$ feltételt, hanem egyedül *csak* a $j = 2^h - 1$ sorszámút, ahol $2^h \le k < 2^{h+1}$. □

**8.25. Megjegyzés.** Ez négyzetes gyorsítás, hiszen minden $k$ esetén $k-1$ helyett csak egyetlen $lnko$ számítást kell elvégeznünk. □

**8.26. Állítás.** *Amennyiben az eredeti algoritmus megtalálta volna $n$ egy valódi osztóját $r = lnko(x_{k_0} - x_{j_0}, n) > 1$ azaz $x_{k_0} \equiv x_{j_0} \pmod r$ alakban, akkor a gyorsított algoritmus is talál megfelelő $k$, $j$ párt, sőt $k \le 4k_0$.*

**Bizonyítás.** $2^h \le k_0 < 2^{h+1}$ és $j_0 < k_0$ esetén $k = 2^{h+1} + (k_0 - j_0) - 1$ és $j = 2^{h+1} - 1$ választással a 8.23. Állítás feltételei teljesülnek, $k \le 4k_0$ nyilvánvaló. ■

**8.27. Példa.** Bontsuk fel az $n = 4087$ számot az $f(x) = x^2 + x + 1$ polinom és az $x_0 = 2$ kezdőérték segítségével.

**8.28. Megoldás.**
$k = 1, h = 0:$ $x_1 = f(2) = 7$, $lnko(x_1 - x_0, n) = lnko(7 - 2, 4087) = 1$,
$k = 2, h = 1:$ $x_2 = f(7) = 57$, $lnko(x_2 - x_1, n) = lnko(57 - 7, 4087) = 1$,
$k = 3, h = 1:$ $x_3 = f(57) = 3307$,
$lnko(x_3 - x_1, n) = lnko(3307 - 7, 4087) = 1$,
$k = 4, h = 2:$ $x_4 = f(3307) \equiv 2745 \pmod{4087}$,
$lnko(x_4 - x_3, n) = lnko(2745 - 3307, 4087) = 1$,
$k = 5, h = 2:$ $x_5 = f(2745) \equiv 1343 \pmod{4087}$,
$lnko(x_5 - x_3, n) = lnko(1343 - 3307, 4087) = 1$,

$k=5, h=2:\quad x_6 = f(1343) \equiv 2626 \pmod{4087},$
$lnko(x_6 - x_3,\, n) = lnko(2626 - 3307,\, 4087) = 1,$
$k=5, h=2:\quad x_7 = f(2626) \equiv 3734 \pmod{4087},$
$lnko(x_7 - x_3,\, n) = lnko(3734 - 3307,\, 4087) = 61.$

Tehát $4087 = 61 \cdot 67$. □

**8.29. Tétel.** *Ha $n \in \mathbb{N}$ páratlan szám, $r \le \sqrt{n}$ valódi osztója, és a $(f, x_0)$ pár kielégíti a 8.22. Állítás feltételeit ($S = \mathbb{Z}_r$, $g = f$ és $s_0 = x_0$ választással), akkor a $\rho$-módszer az $r$ osztót nagy valószínűséggel megtalálja legfeljebb $\mathcal{O}\left(\sqrt[4]{n} \cdot \log^3(n)\right)$ lépésben.*
***Pontosabban:*** *létezik olyan $C \in \mathbb{R}^+$ állandó, hogy bármely $\lambda > 0$ szám esetén: annak a valószínűsége, hogy a $\rho$-módszer az $r$ osztót nem találja meg legfeljebb $C\sqrt{\lambda} \cdot \sqrt[4]{n} \cdot \log^3(n)$ lépésben, kisebb mint $e^{-\lambda}$.* □

**8.30. Megjegyzés.** **(i)** Az $f(x)$ polinomokról csak általában beszéltünk, a 8.22. Állítás sem világos: melyik jobb: $\ell$ nagy vagy kicsi? A polinom persze $r$-től is függ, de a tapasztalat azt mutatja, hogy a népszerű $x^2 + 1$ általában megfelelő.
**(ii)** Például a $\lambda = 9$, $e^{-\lambda} \approx 10^{-4}$ értékek esetén egy átlagos $(f, x_0)$ párra $n$-et majdnem biztosan fel tudjuk bontani.
**(iii)** $f(x) = ax + b$ lineáris vagy az $f(x) = x^2$ tiszta kvadratikus polinomot SOHA sem szabad használni. □

Végül egy olyan módosítást mutatunk, amely a tapasztalat szerint kicsit még az előzőnél is gyorsabb.

**8.31. Algoritmus. (Kétszeres sebesség)** Alkalmazzuk egyszerre az

$$x_{k+1} \equiv f(x_k) \pmod{n} \quad \text{és} \quad x_{2k} \equiv f\left(f(x_{2k-1})\right) \pmod{n}$$

iterációkat, és minden lépésben számítsuk ki $r := lnko(n,\, x_{2k} - x_k)$ értékét!
Ha $r > 1$ akkor nyilván $r$ egy osztója $n$-nek, és $n$ összetett szám. □

**8.32. Példa.** Adott az $n = 246\,733$ természetes szám, $f(x) = x^2 + 1$ és $x_0 = 2$.

| $k$ | $x_k \equiv f(x_{k-1}) \pmod{n}$ | $x_{2k} \equiv f\left(f(x_{2k-1})\right) \pmod{n}$ | $lnko(n,\, x_{2k} - x_k)$ |
|---|---|---|---|
| 0 | 2 | 2 | – |
| 1 | 5 | 26 | 1 |
| 2 | 26 | 211597 | 1 |
| 3 | 677 | 126543 | 1 |
| 4 | 211597 | 99653 | 1 |
| 5 | 133298 | 225011 | 1 |
| 6 | 126543 | 28771 | 1 |
| 7 | 159150 | 90806 | 1 |
| 8 | 99653 | 86408 | 1 |
| 9 | 210626 | 222422 | 983 |

Tehát 983 az egyik osztója $n = 246733$-nak. Az osztást elvégezve megkapjuk $n$ egy faktorizációját: $246733 = 983 \cdot 251$. □

## 8.6. Az AKS algoritmus

Mint a fejezet elején említettük: *Agrawal, Kayal* és *Saxena* 2001-ben feltalált algoritmusa („AKS-teszt") áttörést hozott a *prímtesztelés* problémára: polinomiálisan gyors, 100% biztonságos, determinisztikus. Magát az Algoritmust nem tudjuk bemutatni, mert több matematikai előismeretet követel mint jelen könyvünk, bonyolult. „Kis" számokra a régebbi (könyvünkben ismertetett) algoritmusok jól használhatók.

Az Algoritmus [AKS]-ben jelent meg először. Most csak vázlatosan ismertetjük alapgondolatait.

A teszt alapötlete az, hogy számok helyett polinomokkal dolgozunk. Ha az $n$ számról szeretnénk tudni, hogy prím-e, akkor vizsgáljuk az $f(x) = x^n - a$, $g(x) = (x-a)^n$ polinomokat! Ha $n$ (páratlan) prím, akkor a binomiális tétel és a binomiális együtthatók tulajdonságai miatt a $g(x)$ polinom $(x^n - a^n)$-tól csak $n$-nel osztható tagokban tér el.

Konkrét $x$ és $a$ értékekre kiszámolni a polinomok értékét, majd összehasonlítani $n$-es maradékaikat továbbra sem lenne biztos módszer. Biztonságos, de nagyon időigényes eljárás lenne kiszámolni a polinomokat és együtthatónként $(\mathrm{mod}\ n)$ összevetni egyenlőségüket. Köztes, gyors és ugyanakkor biztonságos módszer a két polinomnak bizonyos polinomokkal vett maradékait összehasonlítani. Ha ugyanis egyenlők a polinomok, akkor bármilyen polinommal vett osztási maradékaik is egyenlők. Alkalmas $(x^r - 1)$ alakú polinomot választani, mert ezzel nagyon könnyű osztani: ilyenkor úgy kell számolni, mintha $(x^r - 1)$ nulla lenne, azaz $x^r$ helyébe mindenhol 1-et kell helyettesíteni. Kiderült, hogy megfelelő olyan $r$ prímet venni, amelynek értéke nagyságrendileg $log^6 n$, és amelyre $r - 1$-nek van egy alkalmas tulajdonságú nagy prímosztója. Ilyen esetben az a szerencse, hogy összetett $n$ szám esetén a kapott maradék-polinomok rendkívül kevés $a$-ra lesznek egyenlők: ha $n \approx 10^{100}$, akkor csak néhány száz kivétel lehet. Elég a maradékban a helyébe behelyettesíteni az első néhány száz értéket, és ellenőrizni az $f(x)$-ből illetve $g(x)$-ből származó értékek $n$-es maradékai megegyeznek. Ha mindegyik próbában egyezés van, akkor kizárt, hogy $n$ összetett, ha egyszer is nincs egyezés, akkor $n$ biztosan összetett.

Az algoritmust feltalálása óta többen egyszerűsítették, különböző módosításait fejlesztették ki (*Lenstra, Pomerance, Crandall, Papadopoulos,* stb.)

**Néhány további internet cím:**
HTTP://MATHWORLD.WOLFRAM.COM/AKSPRIMALITYTEST.HTML,
HTTP://WWW.ANSWERS.COM/TOPIC/AKS-PRIMALITY-TEST,
HTTP://WWW.AMS.ORG/BULL/2005-42-01/S0273-0979-04-01037-7/HOME.HTML

# 9. fejezet

# Prímkeresés

Már Bolyai János előtt több évszázaddal majdnem mindegyik matematikus keresett "*prímképletet*", az összes (vagy legalábbis nagy) prím megkeresésére.

Tényleg igaz, hogy nagyméretű prímszámokat érdemes valamilyen képlettel keresnünk: speciális alakú kifejezések prímtesztelését jóval könnyebb eldönteni, mint csak egy véletlenszerűen „bepötyögött" többezerjegyű számot (karaktersorozatot)? Igen, de az sem meglepő, hogy rengeteg elméleti vizsgálat és még több számítógép futásidő kell többmillió jegyű prímek megtalálásához!

Bár több képlettel is sikerrel próbálkozhatunk (azaz találhatunk időről időre nagy prímeket), a legsikeresebb *Mersenne* képlete. Manapság párezer jegyű prímeket találni „semmiség", ld. pl. a HTTP://WWW.MERSENNE.ORG honlapon. Az első ötvenmillió prím listáját pl. a HTTP://PRIMES.UTM.EDU/LISTS/SMALL/MILLIONS/ honlapon is megtaláljuk.

## 9.1. Mersenne-számok

*Marin Mersenne (1588-1648) francia matematikus* javasolta a következő képletet:

**9.1. Definíció.** Legyen $p \in \mathbb{P}$ tetszőleges *prímszám*.
**(i)** A

$$M_p := 2^p - 1$$

alakú számokat **Mersenne-számok**nak hívjuk (akár összetett, akár prím).
**(ii)** Amennyiben $M_p$ prímszám, akkor őt **Mersenne-prím**nek nevezzük. □

**9.2. Megjegyzés.** Könnyen látható, hogy a $2^k - 1$ alakú számok minden $k \neq 2^\ell$ összetett számra *összetettek* a jólismert

$$a^u - b^u = (a-b)\left(a^{u-1} + a^{u-2}b + \cdots + ab^{u-2} + b^{u-1}\right)$$

azonosság miatt, hiszen ekkor, $k = u \cdot v$, $u \geq 3$ esetén

$$(2^v)^u - 1 = (2^v - 1)\left((2^v)^{u-1} + (2^v)^{u-2} + \cdots + 2^v + 1\right). \quad \square$$

Már $p = 11$ esetén sem prím $M_p$ hiszen $M_{11} = 2^{11} - 1 = 23 \cdot 89$. Azonban ez az egyszerű képlet meglepően hatékony: manapság többszázmillió jegyű (!) prímszámokat, többek között a „*prím-rekordokat*" is segítségével találják meg.

Mely $p \in \mathbb{P}$ prímszámokra lesz $M_p$? A XIX.század óta tudjuk a következő prímtesztet:

**9.3. Tétel** (Lucas–Lehmer teszt)**.** *Legyen $p > 2$ tetszőleges prímszám, $M_p := 2^p - 1$ és legyen $(a_n) \subset \mathbb{N}$ a következő sorozat:*

$$a_1 := 4$$
$$a_{n+1} :\equiv (a_n)^2 - 2 \pmod{M_p}.$$

*Ekkor:* $M_p$ pontosan akkor prím *ha*

$$a_{p-1} \equiv 0 \pmod{M_p}. \quad \square \tag{9.1}$$

*(F. Edouard A. Lucas (1842–1891) francia Derrick Henry Lehmer (1905–1991) amerikai matematikusok.)*

E. Lucas az 1870-es években mondta ki a fenti sejtését, több más tesztmódszerrel együtt, a módszerek helyességét D. H. Lehmer és mások igazolták 1930 körül. A fenti 9.3. Tétel bizonyítása sok helyen megtalálható, pl. komplex számok és a $\mathbb{Z}[i]$ halmaz (ld. a 13.3. Definíció a *Függelék*ben) segítségével, vagy pl: Bruce, J. W: *A Really Trivial Proof of the Lucas–Lehmer Test*, Amer. Math. Monthly, 1993 April, 370-371.

**9.4. Megjegyzés. (i)** Adott $p$ prímszámra a Lucas–Lehmer teszt exponenciális idejű (az $a_1, \ldots, a_{p-1}$ sorozatot egyesével végig ki kell számolnunk) – de csak $p$ számjegyeit tekintve, cserébe viszont $2^p$ méretű prímszámot kapunk, no persze csak akkor, ha a (9.1) feltétel teljesül. Ha nem, kezdjük az egészet elölről, egy másik $p$ prímszámmal.

**(ii)** Prímszámot keresni tehát évekig is eltarthat. Ezért is indult útjára az internetes kollektív prímvadászat: a számítógépek kikapcsolása (vagy képernyővédő programok) helyett a szervezők a hálózatba kapcsolt gépek Lucas-teszt futtatását javasolják *többezer dollár* jutalom mellett! Az érdeklődőknek a HTTP://WWW.MERSENNE.ORG és a HTTP://WWW.UTM.EDU/RESEARCH/PRIMES címeket ajánljuk.

Az 1999-ben felfedezett $M_p$ Mersenne-prím a 38-adik a sorban:

$$M(38) = M_{6\ 972\ 593} = 2^{6\ 972\ 593} - 1 \quad (1999).$$

Néhány régebbi felfedezés: $p = 3.021.377$ (1998. január 27), $p = 2.976.221$ (1997. augusztus 24.), $p = 1.398.269$ (1996. november), $p = 859.433$ (1994. január), $p = 216.091$ (1985). Mersenne-prímek a következők is: $p = 2$, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127 (1950-ig $M_{127}$ volt a legnagyobb ismert prímszám), 521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11.213, 19.937, 21.701, 23.209, 44.497, 86.243, 132.049.
A 2008. és 2009. évi csúcstartók: $p = 37\ 156\ 667$ és $p = 43\ 112\ 609$. $\square$

**9.5. Probléma.** *Máig is megoldatlan probléma, hogy van-e végtelen sok $M_p$ alakú prímszám? Az is megoldatlan, hogy van-e végtelen sok összetett közöttük.* $\square$

(Lásd még a 2.12. Példában $M_{\ddot{o}}$-t.)

A Mersenne-féle prímszámok például a tökéletes számok vizsgálatánál bukkannak fel:

**9.6. Definíció.** Az $n \in \mathbb{N}$ számot **tökéletes szám**nak nevezzük, ha $n$ megegyezik osztóinak összegével. □

**9.7. Tétel** (Euklidesz)**.** *Ha* $m \in \mathbb{N}$ *és* $2^m - 1$ *prímszám, akkor* $2^{m-1} \cdot (2^m - 1)$ *tökéletes szám.* □

Az érdeklődő Olvasóknak még pl. a [LM], [SzI2], [KN] és [FR] műveket ajánljuk (és a 3.46. Állítást).

## 9.2. Fermat-prímek

**9.8. Definíció.** Legyen $n \in \mathbb{N}$ tetszőleges természetes szám. Az

$$F_n := 2^{(2^n)} + 1$$

alakú számokat **Fermat-számok**nak nevezzük,
és **Fermat-prím**nek ha $F_n \in \mathbb{P}$ prímszám. □

**9.9. Megjegyzés.** Könnyen látható, hogy a $2^k + 1$ alakú számok minden $k = 2\ell + 1$ páratlan számra összetettek a jólismert

$$a^{2\ell+1} + b^{2\ell+1} = (a+b)\left(a^{2\ell} - a^{2\ell-1}b + - \cdots - ab^{2\ell-1} + b^{2\ell}\right)$$

azonosság miatt, hiszen ekkor

$$2^{2\ell+1} + 1 = (2+1)\left(2^{2\ell} - 2^{2\ell-1} + - \cdots - 2 + 1\right). \quad \square$$

A Fermat-prímek tehát négyzetszám melletti prímek. Fermat 1650-ben javasolta a képletet prímszámok előállítására, de már Euler igazolta 1732-ben, hogy $n = 5$ esetén

$$F_5 = 2^{2^5} + 1 = 4294\,967\,297$$

nem prím (HF). Jelenlegi tudásunk szerint $n \leq 4$ esetén $F_n$ prím, és $n \geq 5$ esetén minden *megvizsgált* $F_n$ szám összetettnek bizonyult. Mivel az $F_n$ számok szédületesen nagyok, a jelenlegi elméleti és számítástechnikai felszereltséggel reménytelen újabb Fermat-prímeket találni.

**9.10. Probléma.** *Máig* megoldatlan *kérdések: van-e végtelen sok Fermat-prím és van-e végtelen sok* Fermat-összetett szám*?* □

A Fermat-féle prímszámok például a szabályos sokszögek szerkesztésénél játszanak fontos szerepet:

**9.11. Tétel** (Gauss, 1796)**.** *Tetszőleges* $n \in \mathbb{N}$ *természetes számra a szabályos* $n$*-oldalú sokszög akkor és csak akkor szerkeszthető meg körzővel és vonalzóval, ha* $n = 2^s$ *vagy* $n = 2^s \cdot q_1 \cdot \cdots \cdot q_r$ *ahol* $s \in \mathbb{N}$ *és* $q_1, \ldots, q_r$ különböző *Fermat-prímszámok.* □

**9.12. Állítás.** *[W] Az* $F_n = 2^{2^n} + 1$ *Fermat-szám* prím *akkor és csak akkor, ha az* $\dfrac{1}{F_n}$ *szakaszos tizedes tört periódusának hossza pontosan* $2^{2^n}$. □

# 10. fejezet

# Titkosírás nyilvános kulccsal

Igen, kedves Olvasónk: olyan titkosírás következik, amelynek menete (vagyis a kódolás és annak kulcsa) bárki számára *nyilvános*, mégis a levelet csak a címzett tudja elovasni (dekódolni), még maga a levél írója sem!

Mi csak a két legegyszerűbb eljárást mutatjuk be, az elmúlt két évtizedben rengeteg újabb algoritmus látott napvilágot. Az érdeklődőknek például [KN] 95-111. oldalait ajánlhatjuk.

## 10.1. Az RSA-algoritmus

**10.1. Algoritmus. Rivest–Shamir–Adleman algoritmusa (RSA, 1977)**
Az alábbiakban nagyon figyeljünk arra, hogy mely adatok nyilvánosak és melyeket kell titokban tartaniuk a résztvevőknek!
*Az algoritmus leírása 10.3-ig tart.*

Jelöljük a résztvevő személyeket $A$, $B$, $C$, …, $S$, …, $Z$, …-vel.

Az algoritmus leírásához és elemzéséhez hasznosak lesznek a következő jelölések:

**10.2. Jelölés.** $\mathcal{C}_S(x)$ és $\mathcal{D}_S(x)$ jelölje az *x* **szöveg** $S$ **személy** általi **kódolása** illetve **dekódolása** után kapott jelsorozatot. □

A következő **előkészületek**et mindegyik $S$ személy egyedül, titokban végezze:

Először mindenki választ két-két jó nagy (kb. 500-1000 jegyű) prímszámot: $p_A$, $q_A$, $p_B$, $q_B$, …, majd kiszámítják a szorzatukat: $n_A := p_A q_A$, $n_B := p_B q_B$, ….

Az $n$ számokat mindenki nyilvánosságra hozza *„modulus”* elnevezéssel (és saját nevével, elérhetőségével együtt), de a $p$, $q$ prímszámokat természetesen nem! (Ha a személyt nem kell feltüntetnünk, akkor nevét nem írjuk a számok indexébe.)

Felhívjuk a figyelmet, hogy Fermat prímfelbontási algoritmusa (ld.a 7.2. ”*Fermat algoritmusa*” alfejezetben) hatékony akkor, ha $n$ két közeli prím szorzata, tehát titkosírásnál ilyent sem szabad választanunk. Azonban $p-1$ és $q-1$-nek se legyenek kis prímosztói, mert ekkor *Pollard* „$p-1$-algoritmusa” (könyvünkben mi nem tárgyaljuk) az $n$ szám prímfelbontását is könnyen megadja.

Ezután (még mindig titokban) legyen $s := \varphi(n) = (p-1)(q-1)$ (vagyis $s_A$, $s_B$, …). Keressünk továbbá (egyénileg) olyan $e$ számot, amely relatív prím $s$-hez: mondjuk

próbálkozzunk $e = \frac{s}{2}$-vel, ha pedig nem relatív prím $s$-hez, akkor próbálkozzunk tovább az $\frac{s}{2}+1$, $\frac{s}{2}+2$, ... számokkal. (Két szám relatív prím volta könnyen és gyorsan eldönthető azt Euklideszi algoritmussal, Dirichlet 3.47. Tétele és a tapasztalat szerint pedig a próbálkozások elég hamar sikerrel járnak: találunk $s$-hez relatív prím $e$ számot.) Az $e_A$,$e_B$, ... számokat nyilvánosságra hozzák *„nyilvános kulcs”* elnevezéssel.

Végül (mindenki külön, titokban) megkapja az $f$ titkos *„megoldókulcsot”* az

$$e \cdot f \equiv 1 \pmod{s} \quad \text{azaz} \quad ef - sy = 1 \tag{10.1}$$

Diophantikus egyenletből. **Ezt** a (saját) megoldókulcsot – az $f_A$, $f_B$, ... számokat – kell jól elrejtenie mindenkinek, a többi számra (tehát $p$, $q$, $s$-re) már nem lesz szükségünk, de a többiek tudomására sem juthat (pl. „elégetjük” őket).

A titkosírás menete (a protokoll) a következő:

Ha a neveket, elérhetőségeket és az $n$, $e$ számokat nyilvánosságra hoztuk, akkor bárki írhat bárkinek titkosított levelet (amit csak a címzett tud elovasni, még a levél írója sem) – még akkor is, ha előtte nem is hallottak egymásról!

Mondjuk, $B$ szeretne írni $A$-nak. Megírt üzenetét kis részekre bontja és egyesével kódolja őket, egy rész legyen $k < n_A$ egész szám. Az $n_A$ és $e_A$ adatok alapján a $k = k_i$ üzenet $\mathcal{C}_A$-kódja legyen egyszerűen

$$K := \mathcal{C}_A(k) :\equiv k^{e_A} \pmod{n_A} \tag{10.2}$$

a 6.6. *„Nagy kitevőjű hatványozás”* alfejezetben tanult módon. Később megvizsgáljuk, hogy a $K$ üzenet $f_A$ nélkül nem törhető fel (hiába ismeri mindenki $n_A$ és $e_A$ értékét).

$f_A$ birtokában azonban $A$ könnyen elovashatja az üzenetet: legyen

$$\mathcal{D}_A(K) :\equiv K^{f_A} \pmod{n_A}. \tag{10.3}$$

A

$$K^{f_A} \equiv (k^{e_A})^{f_A} = k^{e_A \cdot f_A} = k^{sy+1} = (k^s)^y \cdot k \equiv k \pmod{n_A} \tag{10.4}$$

azonosság szerint elég $A$-nak $K$-t az $f_A$ hatványra emelnie $\pmod{n_A}$:

$$\boxed{B : k \xrightarrow{\mathcal{C}_A} K \ldots \longrightarrow \ldots K \xrightarrow{\mathcal{D}_A} k : A}$$

**10.3. Összegzés. RSA-algoritmus vége.** □

Nem is olyan bonyolult algoritmus – mindössze csak a (10.2) és a lényegében ugyanaz (10.3) képleteket használjuk. A [SzI1] Feladatgyűjtemény 46–48. ill. 112–114. oldalain sok kidolgozott gyakorló feladatot találunk.

Az RSA algoritmus egy sikeres feltörési kísérletének történetét a 10.17 példa kitűzésénél meséljük el, a feladat megoldását 10.23-ben ismertetjük.

Most alaposabban megvizsgáljuk az RSA algoritmust!

**10.4. Állítás.** *$\mathcal{C}_S\left(\mathcal{D}_S\left(x\right)\right) = x$ és $\mathcal{D}_S\left(\mathcal{C}_S\left(y\right)\right) = y$ minden $x, y \in \mathbb{Z}_{n_S}^*$ számra, tehát a $\mathcal{C}_S$ és $\mathcal{D}_S$ függvények egymás inverzei, vagyis $\mathcal{C}_S$ és $\mathcal{D}_S$: $\mathbb{Z}_{n_S}^* \to \mathbb{Z}_{n_S}^*$ invertálható azaz bijektív függvények.*

***Bizonyítás.*** *A hatványozás $\left(x^f\right)^e = \left(x^e\right)^f$ azonossága és a (10.4) levezetés alapján.* ■

**10.5. Megjegyzés.** Különböző személyek algoritmusai $\mathcal{C}_S$, $\mathcal{D}_Z$, $\mathcal{C}_Z$, $\mathcal{D}_S$ azonban nem keverhetőek egymással, semmilyen sorrendben sem össze! □

**10.6. Megjegyzés. *** 0 *** Feltörhetetlenség**
Az algoritmus egy nehéz feltörési történetét a 10.17 példa kitűzésénél meséljük el. □

**10.7. Megjegyzés. *** 1 *** Előkészületek**
A 10.1-ben írt előkészületek minden modern algoritmusnál megszokottak.
Most inkább azt emelnénk ki, hogy amikor $B$ szeretne írni $A$-nak, előtte e két embernek *nem* kell előzetesen *semmiben* megállapodniuk, általában nem is ismerik egymást! (Tudomásunk szerint a biztonsági https honlapokkal való kapcsolat során hasonló információközlés történik.) □

**10.8. Megjegyzés. *** 2 *** Gazdaságosság**
$t$ résztvevő személy esetén nem kell $\binom{t}{2} = \frac{t\cdot(t-1)}{2}$ külön megállapodás a személyek (párok) között csak $t$ db, sőt a titkosírások megállapodásainak nem kell titkosnak lenniük, hiszen a kódolási kulcsok nyilvánosak. □

**10.9. Megjegyzés. *** 3 *** Aláírás hitelesítése**

Igen, a nyilvános kulcs ellenére még az aláírás (levél) is "*hitelesíthető*" az algoritmussal. A fentiek alapján ugyanis könnyen elképzelhető, hogy a fenti $K$ kódolt üzenetet $E$ készítette és küldte el $A$-nak "*írta*: $B$" aláírással. Hogy ezt $B$ elkerülje, leveléhez a következőt csatolja:

Választania kell $B$-nek egy *teljesen véletlen* (eddig és ezután sem használt), nem túl rövid jelsorozatot (pl. neve + dátum másodpercre pontosan + pár véletlen karakter), jelöljük ezt $\ell$-lel. Először $\ell$-et a szokásos módon kódolja: $L :\equiv \ell^{e_A} \pmod{n_A}$ és elküldi $A$-nak:

$$\boxed{B : \ell \xrightarrow{\mathcal{C}_A} L \ldots \longrightarrow \ldots L \xrightarrow{\mathcal{D}_A} \ell : A}$$

Hogy $B$ saját magát igazolja: $f_B$-t azaz $\mathcal{D}_B$-t fogja használni, de természetesen nem mutathatja meg senkinek. Tehát kiszámolja a következőket:

$$\lambda := \mathcal{D}_B\left(\ell\right) \equiv \ell^{f_B} \pmod{n_B}, \quad \Lambda := \mathcal{C}_A\left(\lambda\right) \equiv \lambda^{e_A} \pmod{n_A}$$

és $\Lambda$-t küldi el $A$-nak.

$A$ természetesen el tudja olvasni $\Lambda$-t: $\lambda = \mathcal{D}_A\left(\Lambda\right) \equiv \Lambda^{f_A} \pmod{n_A}$ és $\ell = \mathcal{C}_B\left(\lambda\right) \equiv \lambda^{e_B} \pmod{n_B}$. Végül $A$ összehasonlítja a kétféle módon megkapott $\ell$ üzenetet: $B$ aláírását akkor tekintheti hitelesnek, ha ez a két üzenet megegyezik:

$$\boxed{B : \ell \xrightarrow{\mathcal{D}_B} \lambda \xrightarrow{\mathcal{C}_A} \Lambda \ldots \longrightarrow \ldots \Lambda \xrightarrow{\mathcal{D}_A} \lambda \xrightarrow{\mathcal{C}_B} \ell : A}$$

Azt kell még elhinnünk, hogy $\Lambda$-t *csak* $B$ tudja kiszámolni, hiszen ehhez $f_B$ ismerete szükséges. Matematikusok ezt a kérdést alaposabban megvizsgálták.

Nem muszáj, hogy a szöveg lényegi része és a fenti (hiteles) aláírás egyetlen, szétválaszthatatlan üzenetben legyenek, hiszen mindkettőt ugyanaz az $(n, e)$ pár kódolja, és esetleg az üzenet darabjai valamilyen, a számítástechnikában szokásos módszerrel hivatkozzanak egymásra.

A fenti hitelesítés alkalmazható nem csak személyek azonosítására, hanem tárgyak, (banki, számítógépes vagy egyéb) kódok igazolására is: a tárgyat / kódot birtokló személy igazolni tudja, hogy a kód birtokában van anélkül, hogy az igazolás során bárki (személy vagy számítógép) a kódot megismerhetné vagy ellopná. A titkos $f_B$ kódot persze $B$-nek használnia kell de nem megmutatnia: mivel *csak* a (10.3) számítás *végeredménye* kell, ezért saját számológépén vagy „fejben” is számolhat.

Hasonló módon tudja bárki a saját kódrendszerét nyilvántartásba vetetni saját személyének igazolása után, vagy kódot cserélni a régi vagy az új kód begépelése *nélkül* (amikoris a kód másik gép vagy személy tudomására juthatna). □

**10.10. Megjegyzés. *** 4 *** Megrendelés bizonyítása**
Tételezzük fel, hogy $A$-nak kell igazolnia egy *harmadik* személy (pl. bíróság) felé, hogy az eredetileg $\ell$ tartalmú levelet valóban $B$ írta (ez lényegében az előző pont megfordítása).
$A$ nem használhatja fel sem $\mathcal{D}_A$-t sem $\mathcal{D}_B$-t. Azonban egyszerűen átadja a harmadik személynek $\Lambda$-t és $\lambda$-t ($\ell$-et már felesleges átadnia, $\ell$, $\Lambda$ és $\lambda$ az előző pontban leírt kódok).
A harmadik személy (bíró) ellenőrzi, hogy:
**i)** $\Lambda = \mathcal{C}_A(\lambda)$, vagyis az üzenetet valóban $A$ kapta, és
**ii)** $\ell = \mathbf{C}_B(\lambda)$ vagyis az üzenetet valóban $B$ írta,
és persze elolvassa az $\ell$ üzenet tartalmát.
A fentiek segítségével $A$ sikeresen igazolni tudja, hogy az eredetileg $\ell$ tartalmú levelet valóban $B$ írta. □

**10.11. Összegzés.** Vegyük észre, hogy a fenti 10.6–10.10 hasznos tulajdonságok *minden* olyan nyilvános kulcsú *titkos*írásra érvényesek, amelyre a 10.4 Állítás teljesül. □

## 10.1.1. Példák

Sajnos a különböző példák különböző ABC-ket használnak, ezért alább ismertetjük a használt ABC-ket, valamint minden feladatban megadjuk a példában használt ABC betűszámát (26, 30 vagy 35).

**26-betűs ABC:** 01=A, 02=B, 03=C, 04=D, 05=E, 06=F, 07=G, 08=H, 09=I, 10=J, 11=K, 12=L, 13=M, 14=N, 15=O, 16=P, 17=Q, 18=R, 19=S, 20=T, 21=U, 22=V, 23=W, 24=X, 25=Y, **26**=Z.

**30-betűs ABC:** 01=A, 02=Á, 03=B, 04=C, 05=D, 06=E, 07=É, 08=F, 09=G, 10=H, 11=I, 12=J, 13=K, 14=L, 15=M, 16=N, 17=O, 18=Ö, 19=P, 20=Q, 21=R, 22=S, 23=T, 24=U, 25=Ü, 26=V, 27=W, 28=X, 29=Y, **30**=Z.

**35-betűs ABC:** 01=A, 02=Á, 03=B, 04=C, 05=D, 06=E, 07=É, 08=F, 09=G, 10=H, 11=I, 12=í, 13=J, 14=K, 15=L, 16=M, 17=N, 18=O, 19=Ó, 20=Ö, 21=Ő, 22=P, 23=Q, 24=R, 25=S, 26=T, 27=U, 28=Ú, 29=Ü, 30=Ű, 31=V, 32=W, 33=X, 34=Y, **35**=Z.

00 = szóköz mindig, a rövid üzenetek *elejét* 0-val töltjük fel.

**10.12. Példa.** **a)** Kódolja a „Wir treffen uns am Samstag” [*Találkozzunk szombaton*] üzenetet, ha $n = 55$ és $e = 27$ (26 betűs ABC).
**b)** Dekódolja a 24, 14, 34, 51, 05 RSA üzenetet, ha $n = 55$ és $f = 17$ (35 betűs ABC).
**c)** Dekódolja a 10, 62, 64, 34, 62, 60 RSA üzenetet, ha $n = 77$ és $f = 7$ (35 betűs ABC).

**10.13. Példa.** Adottak a $p = 269$ és $q = 241$ prímszámok és az $e = 53201$ nyilvános kulcs.
**a)** Számolja ki $s = \varphi(n)$ értékét,
**b)** ellenőrizze, hogy $e$ és $s$ relatív prímek, majd számolja ki $f$ értékét,
**c)** kódolja az $x = 48055$ üzenetet,
**d)** dekódolja az előbb kapott titkos üzenetet (azaz ellenőrizze a fenti számításokat),
**e)** kódolja a „HELLO” = 0008 0512 1215 üzenetet (26 betűs ABC),
**f)** dekódolja a 36376 28210 53334 üzenetet.

**10.14. Példa.** Tegyük fel, hogy a mi kódrendszerünk p=23, q=37, n=851, s=792, e=13, f=61, egy társunké p=29, q=31, n=899, s=80, e=29, f=29. Hitelesítsük aláírásunkat részére a „ZSEBSZÁMOLÓGÉP” szöveggel (35 betűs ABC).

**10.15. Példa.** Legyenek $n = 49\ 891\ 381$, $e = 209$, míg $f$, $p$, $q$ és $s$ titkosak, használjuk a 30 betűs ABC-t.
**a)** Kódolja az „ANNA ÖRÖK” = 00000001 16160100 18211813 üzenetet.
**b)** Kódoljuk az „OLVASD EL” üzenetet (26 betűs ABC).
**c)** Ellenőrizze az $z \equiv x^f \equiv 49691150 \pmod n$ aláírás hitelességét.
**d)** Törje fel a kódot ($f$, $p$, $q$, $s$ =?), majd dekódolja az
$y = x^e \equiv 37791786, 01150082, 32137718 \pmod n$ üzenetet.

**10.16. Példa.** Ha
$n =$ 444 113 096 135 661 846 937 = 3 719 977 867 * 119 385 951 211
és $f = 2039$ akkor mennyi $e$ értéke és mennyi az $x = 32$ kódja?

Ha már gyakoroltuk a kódolást/dekódolást, akkor próbáljuk *feltörni* az alábbi titkosírást!

**10.17. Példa. *****:** A *Scientific American* **1977.** augusztusi számában Rivest, Shamir és Adleman tűzték ki az alábbi feladatot és az első megfejtőnek 100$ jutalmat ajánlottak fel (**1994** áprilisában gazdája akadt a 100$-nak):
Törje fel az alábbi kódrendszert: $e = 9007$,

$n = 11438162\ 5757888867\ 6692357799\ 7614661201\ 0218296721\ 2423625625\ 6184293570$
$6935245733\ 8978305971\ 2356395870\ 50589890751\ 4759929002$
$6879543541$ (129 jegyű),

a titkosított üzenet: $K = 9686\ 9613754622\ 06147714092\ 2254355882\ 90575999112$
$4574319874\ 6951209308\ 16298225145\ 70835693147\ 6622883989\ 6280133919$
$9055182994\ 5157815154$ (26 betűs angol ABC)

**10.18. Megjegyzés.** A történet Az [MZ] cikk szerint a következő: $n$ faktorizációja 1994-ben (!) azáltal vált lehetségessé, hogy írtak egy programot, amely a számításokat képes volt sok számítógépre szétosztani s a részeredményeket a központba elküldeni, s több mint 600-an, amikor éppen nem volt szükség számítógépükre, ezt a programot futtatták. A munka így 8 hónapig tartott. A befutott részeredmények egy $569466 \times 524338$ mátrixot alkottak, amelyet Gauss-féle eliminációval $188614 \times 188160$-ra csökkentettek. Ennek alapján a faktorizáció 16K MasPar P-1-es gépen 45 óráig tartott- Ez az első eset, hogy sikerült RSA kódban írt szöveget feltörni; mint láthatjuk, elég szép munka volt.

A fenti feladat végeredményét 10.23-nél ismertetjük. □

További gyakorló feladatok és megoldásuk találhatók még a szerző honlapján:
HTTP://MATH.UNI-PANNON.HU/~SZALKAI/RSA-FELADATOK.TXT,
amelyekhez használjuk a HATVMODDD.EXE progamot.

## 10.1.2. Megoldások

**10.19. Megoldás** (10.12. Példa)**. a)** „Wir treffen uns am Samstag” üzenet kódolva: = 12 04 17 00 15 17 25 41 41 25 09 00 21 09 24 00 01 07 00 24 01 07 24 15 01 28.

**b)** $24^f \equiv 24^{17} \equiv 29 =$Ü $(\mathrm{mod}\, 55)$, ... s.í.t., az üzenet: *ÜGYES*.

**c)** $10^f \equiv 10^7 \equiv 10 =$H $(\mathrm{mod}\, 77)$, ... s.í.t., az üzenet: *HELYES*. □

**10.20. Megoldás** (10.13. Példa)**. a)** $n = pq = 64829$, $s = \varphi(n) = 268 \cdot 204 = 64320$,

**b)** az $ef - sy = 1$, azaz $53201 \cdot f - 64320 \cdot y = 1$ Diophantikus egyenletet kell megoldanunk: $f = 28721$,

**c)** $y \equiv x^e \pmod n$ azaz $y \equiv 48055^{53201} \equiv 61606 \pmod{64829}$,

**d)** $x \equiv y^f \pmod n$ azaz $x \equiv 61606^{28721} \equiv 48055 \pmod{64829}$,

**e)** $8^{53201} \equiv 13745$, $512^{53201} \equiv 57388$ és $1215^{53201} \equiv 18638 \pmod{64829}$, vagyis a „HELLO” üzenet kódolva = 0008 0512 1215,

**f)** $36376^{28721} \equiv 16$, $28210^{28721} \equiv 918$, $1519^{28721} \equiv 53334 \pmod{64829}$, vagyis a kódolt üzenet: 0016 0918 1519 = „PIROS” □

**10.21. Megoldás** (10.15. Példa)**. a)** $00000001^{209} \equiv 00000001$, $16160100^{209} \equiv 00022271 \pmod n$, $18211813^{209} \equiv 47610329 \pmod n$, tehát a kódolt üzenet = 00000001 00022271 47610329 (eml: $n = 49\,891\,381$).

**b)** „OLVASD EL” = 15 12 22 01 19 04 00 05 12 amit 8 hosszú részekre tördelve $k_1 = 1512220$, $k_2 = 11904000$ és $k_3 = 512$. Ekkor

$k_1^e = 1512220^{209} \equiv 11812012 \pmod{49,891,381}$,

$k_2^e = 11904000^{209} \equiv 4882790 \pmod{49,891,381}$,

$k_3^e = 512^{209} \equiv 42839442 \pmod{49,891,381}$,

vagyis a kódolt üzenet: 11812012 4882790 42839442.

**c)** $z^e \equiv 49691150^{209} \equiv 19211115 \pmod n$ és 19 21 11 15 =„PRIM” értelmes üzenet.

**d)** $n = 49891381 = 6091 \cdot 8191 = p \cdot q$,

$s = (p-1)\cdot(q-1) = 49877100$,

$f = 4056989 = 11110111100111100111101^{(BIN}$;

így $(y_1)^f \equiv 37791786^{4056989} \equiv$ 22 30 17 14 =„SZOL” $\pmod n$,

$(y_2)^f \equiv 01150082^{4056989} \equiv$ 11 05 01 21 =„IDAR” $\pmod n$,

$(y_3)^f \equiv 32137718^{4056989} \equiv$ 11 23 02 22 =„ITÁS” $\pmod n$, azaz a feltört üzenet: „SZOLIDARITÁS”. □

**10.22. Megoldás** (10.16. Példa)**.** Ha
$n =$ 444 113 096 135 661 846 937 = 3 719 977 867 * 119 385 951 211 és $f = 2039$, akkor
e= 217 809 267 294 044 099 és $x = 32$ kódja

$y \equiv x^e \equiv$ 316 326 629 379 980 725 998 $\pmod n$. □

**10.23. Megoldás** (10.17. Példa)**.**
$n =$ 3490 5295108476 5094914784 9619903898 1334177646 3849338784 3990820577∗
∗32769 1329932667 0954996198 8190834461 4131776429 6799294253 9798288533.
A „titkos” kitevő:
$f =$ 106 69861436857 8024442868 7713289201 54780709906 63393786280 1226224496 63106312591 17744708733 4016859746 23065539685 4451327710 9053606095.
A hatványozás után az eredeti, rejtjelezett üzenet:
$k =$ 20 08 05 00 13 01 07 09 03 00 23 15 18 04 19 00 01 18 05 00 19 17 21 05 01 13 09 19 08 00 15 19 19 09 06 18 01 07 05
= ” THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE”
(„A VARÁZSSZÓ A KÉNYESGYOMRÚ HALÁSZSAS”). □

## 10.2. A hátizsák algoritmus

Az algoritmus „lelke” a következő (nehéz) probléma:

**10.24. Probléma.** ***Általános hátizsák probléma:*** *Adottak tetszőleges* $m_1, \ldots, m_k$ *és* $M$ *pozitív valós számok (*$k \in \mathbb{N}$ *szintén tetszőleges, nagy szám), és keresendő az alábbi egyenlet*

$$M := \sum_{i=1}^{k} \varepsilon_i \cdot m_i \tag{10.5}$$

*összes* $0-1$ *megoldása, azaz* $\varepsilon_i \in \{0,1\}$ *minden* $i = 1, 2, \ldots, k$ *esetén.* □

A (hivatalos) elnevezést az egyik népszerű „alkalmazás” magyarázza: $M$ teherbírású hátizsákunkba mely tárgyakat tegyük / *ne* tegyük be, melyek tömegei $m_1, \ldots, m_k$, és a zsákot teljesen ki kell használnunk.

Gyors (polinomiális) algoritmus nem ismert az általános kérdés megoldására, mindegyik ismert algoritmus lényegében az összes esetet végigpróbálgatja, vagyis $\mathcal{O}\left(2^k\right)$ idejű. Sőt, jól ismert tétel, hogy az általános hátizsák probléma *NP-teljes* (az NP -teljességet az 1. ”*Algoritmusok sebessége*” fejezetben definiáltuk).

Van azonban a fenti problémának egy *nagyon könnyen* megoldható változata (”*szupernövekvő hátizsák probléma*”), és a titkosírás ezen a kettősségen alapul, de előtte egy segéd definíció kell.

**10.25. Definíció.** Az $(m_1, m_2, \ldots, m_k) \subset \mathbb{R}$ sorozatot **szupernövekvő**-nek nevezzük, ha minden eleme nagyobb mint az összes őt megelőző elem összege, azaz bármely $i = 2, \ldots, k$ esetén

$$m_i > \sum_{j=1}^{i-1} m_j \quad . \quad \square$$

**10.26. Probléma.** ***szupernövekvő hátizsák probléma:*** *Ugyanaz, mint az általános hátizsák probléma, csak az* $(m_1, m_2, \ldots, m_k) \subset \mathbb{R}$ *sorozat szupernövekvő.* $\square$

**10.27. Tétel.** *A szupernövekvő hátizsák probléma lineáris időben megoldható, és a (10.5) egyenletnek legfeljebb csak egy megoldása van.*

**10.28. Algoritmus.** és egy Bizonyítás is.
Az $m_i$ tömegű „csomagokat” csökkenő sorrendben próbáljuk betenni a hátizsákba, vagyis az $\varepsilon_k, \varepsilon_{k-1}, \ldots, \varepsilon_1$ ismeretleneket ebben a sorrendben próbáljuk meghatározni, lehetőleg $\varepsilon_i = 1$ választással úgy, hogy a (10.5) egyenletben egyetlen részösszeg se lépje túl $M$-et. Ha a soron következő $m_i$ belefér még a részösszegbe, vagyis

$$m_i + \sum_{j=i+1}^{k} \varepsilon_j \cdot m_j < M$$

akkor őt kötelező beletennünk, hiszen az utána következő összes többi $m_{i-1}, \ldots, m_1$ számok együttesen is kevesebbet adnak $m_i$-nél, tehát $m_i$ mellőzése esetén biztosan nem telik meg a hátizsák.
Végül tehát vagy megtelik a hátizsák (a problémát megoldottuk), vagy elfogynak a csomagok anélkül, hogy tele tudnánk rakni a hátizsákot (a problémának nincs megoldása).
A megoldás *egyértelműsége* szintén az $(m_1, m_2, \ldots, m_k)$ sorozat szupernövekvő tulajdonságán alapszik. $\square$

Elérkeztünk a titkosírás megismeréséhez.

**10.29. Algoritmus. Merkle–Hellman titkosírás**

Legyenek az elküldendő üzenet egységei $k$-bites számként ábrázolva.

**1)** a felhasználók mindegyike választ magának egy $k+1$-elemű szupernövekvő sorozatot. Legyen például az $A$ felhasználó sorozata $(v_0, v_1, \ldots, v_{k-1}, v_k)$, másként írva $\overrightarrow{v} := (v_0, v_1, \ldots, v_{k-1})$ és $m := v_k$,

**2)** ezután $A$ egyszerű próbálgatással keres egy $m$-hez relatív prím $a$ számot, tehát $lnko(a, m) = 1$,

**3)** majd $A$ meghatározza $a$ multiplikatív inverzét $(\mathrm{mod}\ m)$ (például az Euklideszi algoritmussal), jelölje ezt $b$, azaz $a \cdot b \equiv 1\ (\mathrm{mod}\ m)$,

**4)** Legyen minden $i = 0, 1, \ldots, k-1$ esetén

$$w_i \equiv a \cdot v_i\ (\mathrm{mod}\ m), \tag{10.6}$$

természetesen $0 \leq w_i < m$. A $w_0, w_1, \ldots, w_{k-1}$ sorozat nem lesz szupernövekvő, kivéve néhány extrém esetet.

**A** *nyilvános kulcsa* a $\overrightarrow{w} := (w_0, w_1, \ldots, w_{k-1})$ sorozat lesz; míg **A** *titkos* megoldókulcsai $b$ és $m$.

**5) Üzenetküldés:** Ha valaki szeretne $A$ számára egy üzenetet küldeni, akkor kikeresi a nyilvántartásból $A$ nyilvános kulcsát, $\overrightarrow{w}$ -t. Legyen a titkosítandó üzenet $\overrightarrow{\varepsilon} = (\varepsilon_{k-1}, \varepsilon_{k-2}, \ldots, \varepsilon_0)$ ($\varepsilon_i = 0$ vagy $\varepsilon_i = 1$) amelyből a nyilvános kulcs segítségével elkészítjük a

$$F := \sum_{i=0}^{k-1} \varepsilon_i \cdot w_i \tag{10.7}$$

kódolt üzenetet, $F < km$. (Ez $\overrightarrow{\varepsilon}$ és $\overrightarrow{v}$ skaláris szorzata, de figyeljünk az indexek fordított sorrendjére $\overrightarrow{w}$ és $\overrightarrow{\varepsilon}$-ben, a továbbiakban pedig $H$-nál $\overrightarrow{\varepsilon}$ és $\overrightarrow{v}$-ben!)

Ha a titkos $F$ üzenet egy illetéktelen birtokába jutna, akkor az illetéktelennek $\overrightarrow{w}$ ismeretében az $\varepsilon_i$ (10.7) egyenletet kellene megoldania, az pedig általában egy általános hátizsák probléma. Vagyis nem kell attól félnünk, hogy illetéktelenek elolvashatják levelünket.

**6.) Az üzenet megfejtése (dekódolás):** Miután $A$ megkapja az $F$ titkos üzenetet, először $b$ és $m$ ismeretében kiszámítja a

$$H :\equiv b \cdot F\ \ (\mathrm{mod}\ m)$$

számot. Mivel

$$H = \sum_{i=0}^{k-1} \varepsilon_i\,(bw_i) \equiv \sum_{i=0}^{k-1} \varepsilon_i\,(bav_i) \equiv \sum_{i=0}^{k-1} \varepsilon_i v_i\ \ (\mathrm{mod}\ m)\,,$$

ezért $A$-nak csak a fenti szupernövekvő hátizsák problémát kell megoldania. Mivel $m$ nagyobb, mint az összes $v_i$ együttható összege, ezért a fenti kongruencia egyetlen gyöke éppen az eredeti $\overrightarrow{\varepsilon}$ üzenet.

**Hátizsák algoritmus vége.** □

**10.30. Példa.** Legyen $k = 5$, $\overrightarrow{v} := (2,\ 3,\ 7,\ 15,\ 31)$, $m = 61$. Legyen $a = 17$, ekkor $b = 18$ és (10.6) miatt $\overrightarrow{w} = (34,\ 51,\ 58,\ 11,\ 39) \pmod{61}$. Legyen az elküldendő üzenet $\overrightarrow{\varepsilon} = (10110)$. Ekkor

$$F = 0 \cdot 34 + 1 \cdot 51 + 1 \cdot 58 + 0 \cdot 11 + 1 \cdot 39 = 148$$

lesz a titkosított üzenet.

Ha a címzett megkapja $F$-et, akkor először $H$-t számolja ki:

$$H = 148 \cdot 18 \equiv 41 \pmod{61},$$

majd a $H = \sum_{i=0}^{k-1} \varepsilon_i v_i$ szuperhátizsák-egyenletet megoldva:

$$41 = 1 \cdot 31 + 0 \cdot 15 + 1 \cdot 7 + 1 \cdot 3 + 0 \cdot 2$$

megkapja az eredeti üzenetet: $\overrightarrow{\varepsilon} = (10110)$. □

**10.31. Példa.** Legyen $k := 10$, $\overrightarrow{v} := (1, 3, 7, 14, 28, 54, 110, 219, 437, 875, 1750)$ szupernövekvő sorozat, $m := 3527$ prímszám, továbbá $a = 2222$ szám és az $\overrightarrow{\varepsilon} = 666$ titkosítandó szöveg. Számolja ki $b$-t, $\overrightarrow{w}$-t, az $\overrightarrow{\varepsilon}$ szöveget írja át kettes számrendszerbe majd kódolja. A kapott kódolt üzenetet dekódolja!
**Megoldás:** Az Euklideszi algoritmus szerint $lnko\,(3527, 2222) = 1$, tehát $a = 2222$-nek van inverze. A $2222 \cdot b \equiv 1 \pmod{3527}$ kongruencia / Diophantoszi egyenlet megoldása: $b = 3427 \pmod{3527}$.
A $w_i \equiv a \cdot v_i \pmod{m}$ képlet (10.6) alapján
$\overrightarrow{w} = (2222, 3139, 1446, 2892, 2257, 70, 1057, 3419, 1089, 873, 1746)$.
$\overrightarrow{\varepsilon} = 666 = \overline{01010011010}^{(2}$, és azért írtunk a kettes számrendszerbeli alak elejére egy 0-t, hogy mind a tizenegy bit fel legyen tüntetve.
(10.7) alapján a titkosított üzenet $F = \sum_{i=0}^{k-1} \varepsilon_i \cdot w_i = 12580$.
$F$ dekódolása: először $H \equiv b \cdot F = 12580 \cdot 3427 \equiv 1139 \pmod{3527}$, majd $H$-t kell a szupernövekvő $\overrightarrow{v}$ sorozat elemeinek összegeként felírni, és a kapott 0-1 együtthatók adják a keresett szöveget:

$$1139 = 875+219+28+14+3 = 1\cdot v_9+0\cdot v_8+1\cdot v_7+0\cdot v_6+0\cdot v_5+1\cdot v_4+1\cdot v_3+0\cdot v_2+1\cdot v_1+0\cdot v_0$$

ahonnan $\overrightarrow{\varepsilon} = \overline{01010011010}^{(2} = 666$. □

**10.32. Megjegyzés.** *Shamir* 1982-ben megmutatta, hogy bár $\overrightarrow{w}$ nem szupernövekvő, de az üzenet feltöréséhez jól ki lehet használni azt a tényt, hogy szupernövekvő sorozatból származik. Ezért úgy teszik biztonságosabbá a rendszert, hogy az $(m, a)$ pár alkalmazása helyett dupla titkosítást használnak valamely $(m_1, a_1)$ és $(m_2, a_2)$ párokat felhasználva. □

# 11. fejezet

# Bizonyítás nulla információval

A gyakorlati életben (már több ezer éve) számtalanszor felmerül annak igénye, hogy valamit „bebizonyítsunk” anélkül, hogy a „bizonyítékot” átadnánk vagy akár megmutatnánk a másik félnek, vagy akár elővennénk féltve őrzött titkunkat. Nem csak személyazonosságunkat, hanem titkos kódunkat sem tanácsos másik félnek, még a számítógépnek sem megmutatnunk, bebillenytűznünk.

Egy olyan *párbeszéd-előírást* (protokollt) mutatunk, amellyel a válaszadó (**V**) majdnem 100% hitelességgel be tudja *bizonyítani* a kérdezőnek (**K**) hogy birtokában van a titkos kódnak/információnak anélkül, hogy **K** a kód egy részletét is megismerné. Tehát **V** „bizonyítása” valóban „nulla információval” történik, angolul *zero knowledge proof*. Sőt, akár **K**, akár egy harmadik személy hiába hallotta az elhangzott beszélgetést, legközelebb nem tudja kiadni magát **V**-nek, nem tudja utánozni! (Angolul a *Prover* és a *Verifier* elnevezések használatosak.)

A most ismertetendő eljárás *titkossága* azon alapul, hogy a (6.24) kvadratikus kongruenciák nem oldhatók meg polinomiális időben, vagyis nagy $m$ modulusok esetén nem tudjuk belátható időn belül megoldani azokat, amint ezt a 6.8. „*Magasabbrendű kongruenciák*” alfejezetben láttuk.

Az alábbi algoritmust *Csirmaz László* kedves tanáromtól hallottam 1992-ben. Csirmaz László honlapján: *http://www.math-inst.hu/~csirmaz* sok érdekes anyagot találunk.

**11.1. Algoritmus.** (Feiger–Fiat–Shamir)

**V** választ egy nagy (többszázjegyű) $m$ összetett számot (már $m = pq$ két prím szorzata is elég), és rögzít néhány (legalább pár tucat) $r_1, r_2, \ldots, r_k \in \mathbb{Z}_m^*$ számot, mindezeket titokban tartja és megőrzi. (Például a titkos PIN-kódja az $(r_1, r_2, \ldots, r_k)$ sorozat.)

*Nyilvánosságra hozza* azonban a titkos $r_i$ számok négyzeteit $(\mathrm{mod}\ m)$, jelöljük ezeket az értékeket $s_1, s_2, \ldots, s_k$-val, azaz

$$s_i :\equiv (r_i)^2 \pmod{m}. \tag{11.1}$$

**V**-nek tehát be kell bizonyítania valakinek vagy valamely gépnek – ez utóbbit jelöljük **K**-val, hogy birtokában van a titkos $(r_1, r_2, \ldots, r_k)$ kódnak, anélkül, hogy elárulná ezt a kódot. Pontosabban: **V** azt fogja bebizonyítani, hogy a nyilvános $(s_1, s_2, \ldots, s_k)$ sorozathoz *tartozó* titkos $(r_1, r_2, \ldots, r_k)$ kódot ismeri.

**0)** A bizonyítási eljárás elején **V** választ egy újabb (minden alkalommal más és más) $v \in \mathbb{Z}_m^*$ számot, amit szintén titokban tart, és a

$$w :\equiv v^2 \pmod{m} \tag{11.2}$$

számot adja át **K**-nak.

**1)** **K** ellenőrző kérdése $(e_1, e_2, \ldots, e_k)$ alakú ahol $e_i = 0$ vagy $e_i = 1$ az $i = 1, \ldots, k$ indexekre (lehetőleg az $e_i$ számoknak körülbelül a fele 1).

Válaszként **V** (titokban/fejben) kiszámítja a

$$b :\equiv v \cdot \prod_{i=1}^{k} (r_i)^{e_i} \pmod{m} \tag{11.3}$$

értéket (vagyis $v$-t csak azon $r_i$ számokkal szorozza össze, melyekhez tartozó $e_i = 1$) és a végeredmény $b$-t adja át **K**-nak.

**2)** **K** könnyen ellenőrzi a

$$b^2 \stackrel{?}{\equiv} w \cdot \prod_{i=1}^{k} s^{e_i} \pmod{m} \tag{11.4}$$

feltételt, melynek nyilvánvalóan teljesülnie kell – különben **V** „lebukna".

**3)** Ha **K** nagyon biztos akar lenni, akkor a fenti **1)-2)** pontokat többször, mondjuk $t$-szer meg kell ismételnie. Ugyanis tudjuk, hogy minden $s \in \mathbb{Z}_m^*$ maradék *pontosan két* $r \in \mathbb{Z}_m^*$ érték négyzete. Tehát az egyszer lefolytatott teszten egy csaló $0 < \varepsilon < 1$ valószínűséggel átmehet. Azonban $t$ db független teszten már csak $\varepsilon^t$ eséllyel csúszhat át, ami gyakorlatilag $t \approx 200$ értékkel már $10^{-15}$ alá szorítható. ($t \approx 200$ a mai gépek gyorsaságánál észre sem vehető, és kevesebb mint $10^{10}$ társunk él a Földön.)

**11.2. Összegzés.** Kész a bizonyítás (Feiger–Fiat–Shamir eljárás **vége**). □

Miért *nem tudja* legközelebb senki: akár **K**, akár egy harmadik személy „szimulálni" **V**-t, hiszen a nyilvános kódot és a fenti párbeszédet – azaz (11.1), (11.2), (11.3), (11.4)-t mind hallotta – (majdnem) teljes információval rendelkezik? Mint tudjuk: nagy $m$ modulusok esetén a négyzetgyökvonásra nincs gyors algoritmusunk, nem tudjuk (polinomidőben) $b$, $w$ és $(s_1, s_2, \ldots, s_k)$-ből kiszámolni sem $v$-t sem $(r_1, r_2, \ldots, r_k)$-t.

Természetesen rengeteg, nem csak számelméleti algoritmus létezik a "*Bizonyítás nulla információval*" feladatra. Például a [KRSz] könyv 5.4. fejezetében, vagy Csirmaz László honlapján találunk sok anyagot: HTTP://WWW.MATH-INST.HU/~CSIRMAZ/KRIPT/MATTAN.HTML, vagy például a [CsL] és [GC] cikkekben.

# 12. fejezet

# Számítógépes megvalósítások

A XXI.században nem meglepő, hogy a számítógépünkön és az Interneten rengeteg segédeszköz áll rendelkezésünkre, naponta bővülő kínálatban. Prímfelbontás és prímtesztelés témában a következőket ajánlhatjuk (messze nem a teljesség igényével):

Elméleti olvasnivalók módszerekről és eredményekről például: [MGy], [JA], [W], [TV] munkákban.

Az első pár millió db prímszám listája például a
HTTP://PRIMES.UTM.EDU/LISTS/SMALL/MILLIONS/ címen megtalálható.

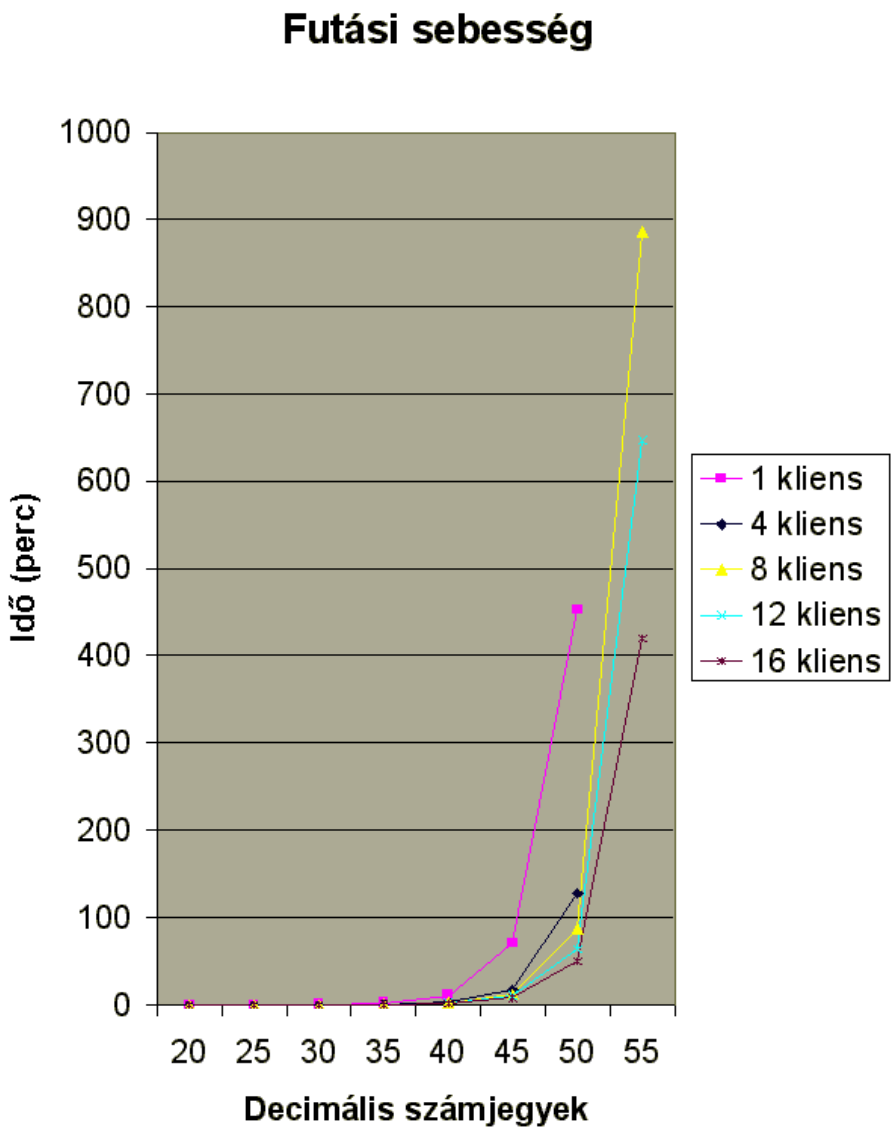
A HTTP://WWW.RSASECURITY.COM/RSALABS/NODE.ASP?ID=2093#RSA576 honlapon versenyt hirdetnek nagy számok faktorizálása témakörben, többezer dollár jutalommal!

A profi és sokoldalú (és méregdrága) Derive ©, Maple © és Mathematica © programokat nem kell bemutatnunk, párszáz jegyű számokkal valós („kivárható") időben boldogulnak – de módszereikről vajmi keveset árulnak el.

Egyetemi hallgatóknak adtam fel házi feladatként 8–20 jegyű számokat faktorizálásra három hónapra, minden segédeszközt használhattak. Saját programjaikkal (és gépeikkel) a nyolcjegyű szám felbontása 1 percig tartott. A Maple V © program *ifactor()* parancsa (saját gépen) 10mp-nél kevesebb idő alatt bontott fel húszjegyű számokat.

A feladott 129 jegyű számmal is elbántak, a következő internet címen található alkalmazás segítségével [W]:
HTTP://WWW.WOLFRAMALPHA.COM/INPUT/?I=FACTOR.
Ez a cím sem árulja el sem az alkalmazott matematikai módszert, sem a működő hardware részleteit.

*Tivolt Viktor* egyetemi hallgató [TV] TDK munkájában (Pannon Egyetem, Veszprém, 2005) a prímfelbontás *párhuzamos* megvalósításait tanulmányozta: 2–16 gépre szétosztva a megoldható feladatok mérete és a sebesség nagymértékben javítható, de pl. kétszer annyi kliens nem biztos, hogy kétszer akkora sebességet eredményez:




Futási idő a különböző nagyságú inputokon többgépes környezetben

# 13. fejezet

# Függelék

## 13.1. Boole-Algebrák

**13.1. Definíció.** A

$$\mathcal{B} = (H, \vee, \wedge, \overline{x}, I, o)$$

struktúra **Boole-algebra (BA)**, ha $H \neq \varnothing$ tetszőleges halmaz, $\vee$, $\wedge$ kétváltozós műveletek, $\overline{x}$ egyváltozós művelet $H$-n, $I$, $o \in H$ konstans elemek, és az alábbi (BA1)–(BA14) tulajdonságok (a Boole-algebra **axiómá**i) teljesülnek:

| | | |
|---|---|---|
| kommutativitás | $A \vee B = B \vee A$ | (BA1) |
| | $A \wedge B = B \wedge A$ | (BA2) |
| asszociativitás | $A \vee (B \vee C) = (A \vee B) \vee C$ | (BA3) |
| | $A \wedge (B \wedge C) = (A \wedge B) \wedge C$ | (BA4) |
| disztributivitás | $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ | (BA5) |
| | $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ | (BA6) |
| elnyelési tulajdonságok | $A \vee (A \wedge B) = A$ | (BA7) |
| | $A \wedge (A \vee B) = A$ | (BA8) |
| $\emptyset$ és $I$ tulajdonságai | $A \vee \overline{A} = I$ | (BA9) |
| | $A \wedge \overline{A} = o$ | (BA10) |
| | $A \vee o = A$ | (BA11) |
| | $A \wedge o = o$ | (BA12) |
| | $A \vee I = I$ | (BA13) |
| | $A \wedge I = A$ | (BA14) |

*$A \vee$ műveletet általában* **konjukció***nak,* $\wedge$-t **diszjunkció***nak, az* $\overline{x}$ *műveletet* **komplementer***nek, míg az* $I, o$ *elemeket* **egység-** *illetve* **nullelem***nek hívjuk.* □

Nem csak a halmazelmélet és a logika műveletei alkotnak Boole-algebrákat: a számelméleti $lnko$ és $lkkt$ is (ld. 3.38. Tétel), az optika „*szubtraktív*" és "*additív*" színkeverése is, és még sok más. A [SzI2] köny 1. fejezetében részletesebben olvashatunk a Boole-Algebrákról.

## 13.2. Polinomok, Euklideszi gyűrűk

Az könyvben tárgyalt fogalmak és összefüggések **(***oszthatóság, prímek, lnko, Euklideszi algoritmus, lineáris Diophantoszi egyenletek, kongruenciák és maradékosztályok, stb.***)** nem csak az egész számok $\mathbb{Z}$ halmazán, hanem sok más gyűrűben (összeadás és szorzás műveletével ellátott halmazon) is léteznek, mint például a polinomoknál, a komplex számok bizonyos részhalmazain (pl. Euler- és Gauss egészek), stb. Ezen vizsgálatok nagy része nem csak elméleti, hanem gyakorlati problémák megoldásához is segítséget nyújt.

Most csak a legalapvetőbb definíciókat és tételeket említjük meg. Az érdeklődők részletesen kidolgozott feladatokat találnak pl. [SzI1] 49–50., 52. és 56-57 oldalain.

**13.2. Definíció. (i)** $\mathbb{Z}[x]$, $\mathbb{R}[x]$ ill. $\mathbb{C}[x]$ jelöli rendre az egész-, valós- ill. komplex együtthatójú (egyismeretlenes) polinomok halmazát.
**(ii)** Egy $p(x)$ polinom **fokszáma** (**grade/degree**) az $x$ legmagasabb előforduló hatványkitevője, jele $\mathbf{gr}(\mathbf{p})$ vagy $\mathbf{d}(\mathbf{p})$ .
**(iii)** az azonosan $c$ értéket felvevő $0$ fokú **konstans** polinomot $\underset{\sim}{c}$ -al jelöljük. Minden $\underset{\sim}{c}$ konstans polinom fokszáma $0$, kivétel a $\underset{\sim}{0}$ polinom fokszáma:
$gr(\underset{\sim}{0}) := -\infty$. $\square$

**13.3. Definíció. (i)** Egy $\alpha \in \mathbb{C}$ szám **$k$-adfokú** ($k \in \mathbb{N}$) **algebrai** szám, ha $\alpha$ gyöke egy $k$-adfokú *valós* együtthatós (azaz $\mathbb{R}[x]$-beli) polinomnak,
**(ii)** $\alpha \in \mathbb{C}$ **algebrai egész,** ha $\alpha$ gyöke egy *egész* együtthatós (azaz $\mathbb{Z}[x]$-beli) polinomnak.
**(iii)** Legyen $\alpha \in \mathbb{C}$ egy tetszőleges *másodfokú* algebrai egész szám, mely gyöke egy

$$\alpha^2 + p\alpha + q = 0 \tag{13.1}$$

egész együtthatós polinomnak. Ekkor legyen

$$\mathbb{Z}[\alpha] := \{a + b \cdot \alpha \mid a, b \in \mathbb{Z}\}.$$

(Természetesen $\mathbb{Z}[\alpha] \subset \mathbb{C}$.)
$\alpha = \sqrt{m}$ esetén (ha $m \in \mathbb{N}$ és *nem* négyzetszám) $\mathbb{Z}[\alpha]$ helyett használatos még a $H_m$ jelölés is:

$$H_m := \mathbb{Z}[\sqrt{m}].$$

**Például:**
$\mathbb{Z}[i]$ elemeit **Gauss-egészek**nek hívjuk,
$\mathbb{Z}[\rho]$ elemeit **Euler-egészek**nek hívjuk,
$\mathbb{Z}[\sqrt{2}]$ elemeit **H-egészek**nek hívjuk,
$\mathbb{Z}[i\sqrt{5}]$ elemeit **L-egészek**nek hívjuk,
.... $\square$

**13.4. Állítás.** $\mathbb{Z}[\alpha]$ *zárt az alapműveletekre (összeadás és szorzás) ha* $\alpha$ másodfokú *algebrai egész szám.*

**Bizonyítás.** Az összeadás és kivonás nyilvánvaló.
$(a+b\alpha)\cdot(c+d\alpha)=(ac-bdq)+(ad+bc-bdp)\,\alpha$ a (13.1) egyenlet felhasználásával, így $\mathbb{Z}[\alpha]$ a szorzásra is zárt. ■

Több számelméleti kérdésre is a $\mathbb{Z}[\alpha]$ halmazok segítségével kaptunk választ, mint például a 3.57. *Nagy Fermat Tétel* $n=3$ és $n=4$ eseteire (Euler és Fermat), valamint Fermat 3.58. „*karácsonyi*" Tételére (Bolyai János).

Az alábbiakban a $(\mathcal{R},+,\cdot)$ struktúra helyére gondoljuk legtöbbször a fenti $\mathbb{Z}[x]$, $\mathbb{R}[x]$, $\mathbb{C}[x]$ vagy $\mathbb{Z}[\alpha]$ halmazok valamelyikét.

**13.5. Definíció. (i)** Legyen $a,b\in\mathcal{R}$ két tetszőleges elem. $a$ **osztója** $b$-nek, vagy $b$ **osztható** $a$-val, jelben

$$a\mid b\,,$$

ha

$$b=a\cdot c$$

valamely $c\in\mathcal{R}$ elemre.

**(ii)** Az $e\in\mathcal{R}$ elem **egység** (≠egységelem!), ha

$$e\mid 1\,.$$

**(iii)** $a,b\in\mathcal{R}$ **asszociáltak (társítottak)**, jelben

$$a\sim b$$

ha

$$a=b\cdot e$$

valamely $e\in\mathcal{R}$ egységre.
(Ekkor szintén $b=a\cdot f$ valamely másik $f\in\mathcal{R}$ egységre.)

**(iv)** Tetszőleges $a\in\mathcal{R}$ **irreducibilis (felbonthatatlan),** ha nincs valódi felbontása, azaz bármely $a=b\cdot c$ $(b,c\in\mathcal{R})$ esetén

$$a=b\cdot c\quad\Longrightarrow\quad b \text{ vagy } c \text{ egység.}$$

A felbontható elemeket **reducibilis**nek nevezzük.
**(v)** $a\in\mathcal{R}$ **prímtulajdonságú** vagy röviden **prím** elem, ha bármely $b,c\in\mathcal{R}$ esetén

$$a\mid b\cdot c\quad\Longrightarrow\quad a\mid b \text{ vagy } a\mid c.$$

□

Az $a\sim b$ *jelölés* ($a$ és $b$ *asszociáltak*) sajnos megegyezik a 2.5. Definícióban bevezetett $f(x)\sim g(x)$ (*aszimptotikusan egyenlő függvények*) jelöléssel, de egészen mást jelölnek!

**13.6. Példa.** Keressük meg $\mathbb{R}[x]$-ben a fenti tulajdonságú elemeket:

Egységek a $\underset{\sim}{c}$ konstans polinomok ($c \neq 0$).

Két polinom pontosan akkor asszociált, ha egy (nemnulla) konstans szorzóban térnek csak el egymástól, például $(2x^2 + 3x - 4) \sim (x^2 + 1.5x - 2)$.
Minden elsőfokú polinom (nyilván) felbonthatatlan, a másodfokúak közül pedig pontosan azok, melyek diszkriminánsa negatív. Az Algebra Alaptétele szerint a legalább harmadfokú polinomok mind reducibilisek.
Az alábbi (általános) tételekből következik majd, hogy $\mathbb{R}[x]$-ben a felbonthatatlan és a prímtulajdonságú elemek ugyanazok. □

Sok kidolgozott gyakorló feladatot találunk még [SzI1]-ben.

Térjünk vissza az $(\mathcal{R}, +, \cdot)$ struktúrák általános vizsgálatához. Az alábbi eredmények csak **integritási tartomány**okra (kommutatív egységelemes gyűrűk) igazak, de ne ijedjünk meg: a fenti $\mathbb{Z}[x]$, $\mathbb{R}[x]$, $\mathbb{C}[x]$ és $\mathbb{Z}[\alpha]$ halmazok mindegyike integritási tartomány.

**13.7. Állítás.** *Tetszőleges $(\mathcal{R}, +, \cdot)$ integritási tartományban: minden prímtulajdonságú elem irreducibilis.*

**Bizonyítás.** Ha $p = bc$ akkor $p \mid bc$, $p$ prímtulajdonsága miatt mondjuk $p \mid b$ vagyis $b = pd$.
Ekkor azonban $p = pdc$ vagyis $dc$ egység és így $c$ is egység tehát a $p = bc$ felbontás nem valódi vagyis $p$ irreducibilis. ■

**13.8. Megjegyzés.** A következtetés visszafelé általában nem igaz, például *bizonyos* $\mathbb{Z}[\alpha]$ struktúrákban. Az alábbiakban ezt a kérdést próbáljuk tisztázni.

**13.9. Definíció.** Az $(\mathcal{R}, +, \cdot)$ *struktúrában teljesül* az **egyértelmű prímfelbontás,** röviden **EPF** *tulajdonsága,* ha bármely $a \in \mathcal{R}$ felbontható irreducibilis elemek szorzatára *lényegében egyértelműen* (sorrendtől és asszociáltaktól eltekintve). □

A fenti elnevezés nagyon hasznos tulajdonságot takar: ha minden elemet **atomok**ra (="*tovább már nem bontható*" elemek, gör.) tudunk bontani, akkor így könnyebben tudjuk az elemek tulajdonságait, a műveleteket vizsgálni.

**13.10. Megjegyzés.** Például a páros számoknál nem egyértelmű a prímfelbontás, pl. $60 = 2 \cdot 30 = 10 \cdot 6$ két különböző felbontás.
$\mathbb{Z}[\sqrt{10}]$-ben sem teljesül az EPF. □

A következő eredmény igazolása már meghaladja könyvünk kereteit:

**13.11. Tétel.** *Egy $(\mathcal{R}, +, \cdot)$ integritási tartományban* pontosan akkor *teljesül az egyértelmű prímfelbontás* ha *minden irreducibilis elem prímtulajdonságú.* □

No, és mikor teljesülnek a fenti Tétel feltételei?

**13.12. Definíció.** Legyen $(\mathcal{R}, +, \cdot)$ tetszőleges.
**(i)** $(\mathcal{R}, +, \cdot)$-ben *elvégezhető* a **maradékos osztás,** ha létezik egy

$$\varphi : \mathcal{R} \longrightarrow \mathbb{N}$$

függvény a következő tulajdonsággal: tetszőleges $a, b \in \mathcal{R}$, $\varphi(b) \neq 0$ elemekhez találhatók olyan $q, r \in \mathcal{R}$ elemek, amelyekre

$$a = bq + r \quad \text{és} \quad \varphi(r) < \varphi(b).$$

($q$ a hányados, $r$ a maradék).
$\varphi()$ helyett néha $||$-t vagy $|||$-t írnak, és "**abszolút érték**" -nek vagy "**normá**" -nak nevezik (hiszen $\varphi$ az elemek „nagyságát" méri ).

**(ii)** Ha $(\mathcal{R}, +, \cdot)$ integritási tartomány és benne elvégezhető a maradékos osztás, akkor $(\mathcal{R}, +, \cdot)$-et **Euklideszi gyűrű**nek nevezzük. □

**13.13. Tétel.** *Minden Euklideszi gyűrűben minden irreducibilis elem prímtulajdonságú.*
***Következésképpen:*** *Euklideszi gyűrűkben teljesül az egyértelmű prímfelbontás.* □

**13.14. Következmény.** *A fenti Tételből, pontosabban a maradékos osztásból következik (végső soron) a 3.6. Számelmélet Alaptétele, ill. az Algebra Alaptételének fele:*
*„Minden egész/valós/komplex együtthatós polinom (vagyis $\mathbb{Q}[x]$ és $\mathbb{R}[x]$ elemei) lényegében egyértelműen (sorrendtől és konstans szorzóktól eltekintve) bontható fel irreducibilis elemek szorzatára".* □

**13.15. Megjegyzés.** **(i)** Például a $\mathbb{Q}[x]$, $\mathbb{R}[x]$, $\mathbb{Z}[i]$, $\mathbb{Z}[i\sqrt{2}]$,$\mathbb{Z}[\rho]$, $\mathbb{Z}[\sqrt{2}]$, $\mathbb{Z}[\sqrt{3}]$, $\mathbb{Z}[\sqrt{6}]$, $\mathbb{Z}[\sqrt{7}]$, $\mathbb{Z}[\sqrt{11}]$, $\mathbb{Z}[\sqrt{19}]$ struktúrák Euklideszi gyűrűk, tehát bennük érvényes az EPF.
**(ii)** A fenti 13.13. Tétel következtetése azonban nem fordítható meg: vannak olyan gyűrűk, amelyek ugyan *nem Euklidesziek*, de bennük mégis érvényes az EPF. Ilyenek például a $\mathbb{Z}[x]$, $\mathbb{Z}[\sqrt{23}]$, $\mathbb{Z}[i\sqrt{3}]$, $\mathbb{Z}[i\sqrt{19}]$, $\mathbb{Z}[i\sqrt{43}]$, $\mathbb{Z}[i\sqrt{67}]$, $\mathbb{Z}[i\sqrt{163}]$ struktúrák. □

Máig **megoldatlan** többek között a következő probléma ([SzM] 22.old.):

**13.16. Probléma.** ***(o)*** *Mely $m \in \mathbb{Z}$ (nem négyzetszám) egész számokra teljesül $\mathbb{Z}[\sqrt{m}]$-ben az egyértelmű prímfelbontás?*
*Az alábbi eredmények ismertek ([SzM]):*

***(i)*** Negatív $m$ *esetén ismert az összes megfelelő $m$ szám:* $\mathbb{Z}[i]$, $\mathbb{Z}[i\sqrt{2}]$, $\mathbb{Z}[i\sqrt{3}]$, $\mathbb{Z}[i\sqrt{7}]$, $\mathbb{Z}[i\sqrt{11}]$, $\mathbb{Z}[i\sqrt{19}]$, $\mathbb{Z}[i\sqrt{43}]$, $\mathbb{Z}[i\sqrt{67}]$, $\mathbb{Z}[i\sqrt{163}]$*-ben* igaz *az EPF (kb. 1970 óta tudjuk biztosan, hogy nincs több megfelelő negatív $m$).*
***(ii)*** Nem *érvényes az EPF $\mathbb{Z}[\sqrt{m}]$-ben minden olyan* pozitív $m \in \mathbb{N}$ *(nem négyzet-) számra, amelyre*

$$4 \mid m - 1\,.$$

***(iii)*** *Ismertek még: $\mathbb{Z}[\sqrt{2}]$, $\mathbb{Z}[\sqrt{3}]$, $\mathbb{Z}[\sqrt{6}]$, $\mathbb{Z}[\sqrt{7}]$, $\mathbb{Z}[\sqrt{11}]$, $\mathbb{Z}[\sqrt{19}]$ és $\mathbb{Z}[\sqrt{23}]$-ben érvényes az EPF, míg $\mathbb{Z}[\sqrt{10}]$-ben nem.* □

**13.17. Megjegyzés.** A fentiek szerint tehát (majdnem) az *egész könyv* Definíciói, Tételei és Algoritmusai igazak ill. használhatók azon fenti $Z[\alpha]$ struktúrákban, amelyekben érvényes az EFP. Ilyenek például:
- lnko és lkkt fogalma és összefüggései,
- Euklidesz algoritmusa lnko keresésére,
- Lineáris Diophantikus egyenletek elmélete és gyakorlata,
- Kínai Maradéktétel és alkalmazásai,
stb. . . . . □

A mellékelt POLIOSZ5.EXE program segítségével gyakorolhatjuk a polinomok maradékos osztását, Euklideszi algoritmust, lnko keresését . . . a $\mathbb{Z}[x]$ gyűrűben.

A jelen fejezetben leírtak nem csak elméletileg, hanem gyakorlati problémáknál is fontosak és hasznosak. A [SzI1] Feladatgyűjteményben sok részletesen kidolgozott feladatot találunk polinomokról és a $\mathbb{Z}[\alpha]$ struktúrákról.

## 13.3. Táblázatok

A 30 000-nél kisebb számok felbontását megtalálhatjuk például a [SzI1] Feladatgyűjtemény Függelékében, az első 50 millió db prímszám listáját például a HTTP://PRIMES.UTM.EDU/LISTS/SMALL/MILLIONS/ címen megtaláljuk.

Alább csak primitív gyökök, hatványaik és számok indexeinek (diszkrét logaritmus) táblázatát mellékeljük. A táblázatok és használatuk leírását elsősorban a 6.7. „*Primitív gyökök és diszkrét logaritmus*” alfejezetben találjuk. Ajánljuk még a 6.71. Példát és a 6.83. Megjegyzést.

| p | g | p | g | p | g | p | g | p | g | p | g |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 113 | 3 | 277 | 5 | 457 | 13 | 643 | 11 | 839 | 11 |
| 3 | 2 | 127 | 3 | 281 | 3 | 461 | 2 | 647 | 5 | 853 | 2 |
| 5 | 2 | 131 | 2 | 283 | 3 | 463 | 3 | 653 | 2 | 857 | 3 |
| 7 | 3 | 137 | 3 | 293 | 2 | 467 | 2 | 659 | 2 | 859 | 2 |
| 11 | 2 | 139 | 2 | 307 | 5 | 479 | 13 | 661 | 2 | 863 | 5 |
| 13 | 2 | 149 | 2 | 311 | 17 | 487 | 3 | 673 | 5 | 877 | 2 |
| 17 | 3 | 151 | 6 | 313 | 10 | 491 | 2 | 677 | 2 | 881 | 3 |
| 19 | 2 | 157 | 5 | 317 | 2 | 499 | 7 | 683 | 5 | 883 | 2 |
| 23 | 5 | 163 | 2 | 331 | 3 | 503 | 5 | 691 | 3 | 887 | 5 |
| 29 | 2 | 167 | 5 | 337 | 10 | 509 | 2 | 701 | 2 | 907 | 2 |
| 31 | 3 | 173 | 2 | 347 | 2 | 521 | 3 | 709 | 2 | 911 | 17 |
| 37 | 2 | 179 | 2 | 349 | 2 | 523 | 2 | 719 | 11 | 919 | 7 |
| 41 | 6 | 181 | 2 | 353 | 3 | 541 | 2 | 727 | 5 | 929 | 3 |
| 43 | 3 | 191 | 19 | 359 | 7 | 547 | 2 | 733 | 6 | 937 | 5 |
| 47 | 5 | 193 | 5 | 367 | 6 | 557 | 2 | 739 | 3 | 941 | 2 |
| 53 | 2 | 197 | 2 | 373 | 2 | 563 | 2 | 743 | 5 | 947 | 2 |
| 59 | 2 | 199 | 3 | 379 | 2 | 569 | 3 | 751 | 3 | 953 | 3 |
| 61 | 2 | 211 | 2 | 383 | 5 | 571 | 3 | 757 | 2 | 967 | 5 |
| 67 | 2 | 223 | 3 | 389 | 2 | 577 | 5 | 761 | 6 | 971 | 6 |
| 71 | 7 | 227 | 2 | 397 | 5 | 587 | 2 | 769 | 11 | 977 | 3 |
| 73 | 5 | 229 | 6 | 401 | 3 | 593 | 3 | 773 | 2 | 983 | 5 |
| 79 | 3 | 233 | 3 | 409 | 21 | 599 | 7 | 787 | 2 | 991 | 6 |
| 83 | 2 | 239 | 7 | 419 | 2 | 601 | 7 | 797 | 2 | 997 | 7 |
| 89 | 3 | 241 | 7 | 421 | 2 | 607 | 3 | 809 | 3 | 1009 | 11 |
| 97 | 5 | 251 | 6 | 431 | 7 | 613 | 2 | 811 | 3 | | |
| 101 | 2 | 257 | 3 | 433 | 5 | 617 | 3 | 821 | 2 | | |
| 103 | 5 | 263 | 5 | 439 | 15 | 619 | 2 | 823 | 3 | | |
| 107 | 2 | 269 | 2 | 443 | 2 | 631 | 3 | 827 | 2 | | |
| 109 | 6 | 271 | 6 | 449 | 3 | 641 | 3 | 829 | 2 | | |

$p = 3$, $g = 2$

| **Szám** | **0** | **1** | **2** |
|---|---|---|---|
| **0** | | 2 | 1 |

| **Ind.** | **0** | **1** | **2** |
|---|---|---|---|
| **0** | | 2 | 1 |

$p = 5$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | | 4 | 1 | 3 | 2 |

| **Ind.** | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | | 2 | 4 | 3 | 1 |

$p = 7$, $g = 3$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| **0** | | 6 | 2 | 1 | 4 | 5 | 3 |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| **0** | | 3 | 2 | 6 | 4 | 5 | 1 |

$p = 11$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 10 | 1 | 8 | 2 | 4 | 9 | 7 | 3 | 6 |
| **1** | 5 | | | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 |
| **1** | 1 | | | | | | | | | |

$p = 13$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 12 | 1 | 4 | 2 | 9 | 5 | 11 | 3 | 8 |
| **1** | 10 | 7 | 6 | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 2 | 4 | 8 | 3 | 6 | 12 | 11 | 9 | 5 |
| **1** | 10 | 7 | 1 | | | | | | | |

$p = 17$, $g = 3$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 16 | 14 | 1 | 12 | 5 | 15 | 11 | 10 | 2 |
| **1** | 3 | 7 | 13 | 4 | 9 | 6 | 8 | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 3 | 9 | 10 | 13 | 5 | 15 | 11 | 16 | 14 |
| **1** | 1 | 8 | 7 | 4 | 12 | 2 | 6 | 1 | | |

$p = 19$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 18 | 1 | 13 | 2 | 16 | 14 | 6 | 3 | 8 |
| **1** | 17 | 12 | 15 | 5 | 7 | 11 | 4 | 10 | 9 | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 2 | 4 | 8 | 16 | 13 | 7 | 14 | 9 | 18 |
| **1** | 17 | 15 | 11 | 3 | 6 | 12 | 5 | 10 | 1 | |

$p = 23$, $g = 5$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 22 | 2 | 16 | 4 | 1 | 18 | 19 | 6 | 10 |
| **1** | 3 | 9 | 20 | 14 | 21 | 17 | 8 | 7 | 12 | 15 |
| **2** | 5 | 13 | 11 | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 5 | 2 | 10 | 4 | 20 | 8 | 17 | 16 | 11 |
| **1** | 9 | 22 | 18 | 21 | 13 | 19 | 3 | 15 | 6 | 7 |
| **2** | 12 | 14 | 1 | | | | | | | |

$p = 29$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 28 | 1 | 5 | 2 | 22 | 6 | 12 | 3 | 10 |
| **1** | 23 | 25 | 7 | 18 | 13 | 27 | 4 | 21 | 11 | 9 |
| **2** | 24 | 17 | 26 | 20 | 8 | 16 | 19 | 15 | 14 | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 2 | 4 | 8 | 16 | 3 | 6 | 12 | 24 | 19 |
| **1** | 9 | 18 | 7 | 14 | 28 | 27 | 25 | 21 | 13 | 26 |
| **2** | 23 | 17 | 5 | 10 | 20 | 11 | 22 | 15 | 1 | |

$p = 31$, $g = 3$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 30 | 24 | 1 | 18 | 20 | 25 | 28 | 12 | 2 |
| **1** | 14 | 23 | 19 | 11 | 22 | 21 | 6 | 7 | 26 | 4 |
| **2** | 8 | 29 | 17 | 27 | 13 | 10 | 5 | 3 | 16 | 9 |
| **3** | 15 | | | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 3 | 9 | 27 | 19 | 26 | 16 | 17 | 20 | 29 |
| **1** | 25 | 13 | 8 | 24 | 10 | 30 | 28 | 22 | 4 | 12 |
| **2** | 5 | 15 | 14 | 11 | 2 | 6 | 18 | 23 | 7 | 21 |
| **3** | 1 | | | | | | | | | |

$p = 37$, $g = 2$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 36 | 1 | 26 | 2 | 23 | 27 | 32 | 3 | 15 |
| **1** | 24 | 30 | 28 | 11 | 33 | 13 | 4 | 7 | 17 | 35 |
| **2** | 25 | 22 | 31 | 15 | 29 | 10 | 12 | 6 | 34 | 21 |
| **3** | 14 | 9 | 5 | 20 | 8 | 19 | 18 | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 2 | 4 | 8 | 16 | 32 | 27 | 17 | 34 | 31 |
| **1** | 25 | 13 | 26 | 15 | 30 | 23 | 9 | 18 | 36 | 35 |
| **2** | 33 | 29 | 21 | 5 | 10 | 20 | 3 | 6 | 12 | 24 |
| **3** | 11 | 22 | 7 | 14 | 28 | 19 | 1 | | | |

$p = 41$, $g = 6$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 40 | 26 | 15 | 12 | 22 | 1 | 39 | 38 | 30 |
| **1** | 8 | 3 | 27 | 31 | 25 | 37 | 24 | 33 | 16 | 9 |
| **2** | 34 | 14 | 29 | 36 | 13 | 4 | 17 | 5 | 11 | 7 |
| **3** | 23 | 28 | 10 | 18 | 19 | 21 | 2 | 32 | 35 | 6 |
| **4** | 20 | | | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 6 | 36 | 11 | 25 | 27 | 39 | 29 | 10 | 19 |
| **1** | 32 | 28 | 4 | 24 | 21 | 3 | 18 | 26 | 33 | 34 |
| **2** | 40 | 35 | 5 | 30 | 16 | 14 | 2 | 12 | 31 | 22 |
| **3** | 9 | 13 | 37 | 17 | 20 | 38 | 23 | 15 | 8 | 7 |
| **4** | 1 | | | | | | | | | |

$p = 43$, $g = 3$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 42 | 27 | 1 | 12 | 25 | 28 | 35 | 39 | 2 |
| **1** | 10 | 30 | 13 | 32 | 20 | 26 | 24 | 38 | 29 | 19 |
| **2** | 37 | 36 | 15 | 16 | 40 | 8 | 17 | 3 | 5 | 41 |
| **3** | 11 | 34 | 9 | 31 | 23 | 18 | 14 | 7 | 4 | 33 |
| **4** | 22 | 6 | 21 | | | | | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 3 | 9 | 27 | 38 | 28 | 41 | 37 | 25 | 32 |
| **1** | 10 | 30 | 4 | 12 | 36 | 22 | 23 | 26 | 35 | 19 |
| **2** | 14 | 42 | 40 | 34 | 16 | 5 | 15 | 2 | 6 | 18 |
| **3** | 11 | 33 | 13 | 39 | 31 | 7 | 21 | 20 | 17 | 8 |
| **4** | 24 | 29 | 1 | | | | | | | |

$p = 47$, $g = 5$

| **Szám** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 46 | 18 | 20 | 36 | 1 | 38 | 32 | 8 | 40 |
| **1** | 19 | 7 | 10 | 11 | 4 | 21 | 26 | 16 | 12 | 45 |
| **2** | 37 | 6 | 25 | 5 | 28 | 2 | 29 | 14 | 22 | 35 |
| **3** | 39 | 3 | 44 | 27 | 34 | 33 | 30 | 42 | 17 | 31 |
| **4** | 9 | 15 | 24 | 13 | 43 | 41 | 23 | | | |

| **Ind.** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | | 5 | 25 | 31 | 14 | 23 | 21 | 11 | 8 | 40 |
| **1** | 12 | 13 | 18 | 43 | 27 | 41 | 17 | 38 | 2 | 10 |
| **2** | 3 | 15 | 28 | 46 | 42 | 22 | 16 | 33 | 24 | 26 |
| **3** | 36 | 39 | 7 | 35 | 34 | 29 | 4 | 20 | 6 | 30 |
| **4** | 9 | 45 | 37 | 44 | 32 | 19 | 1 | | | |

# Irodalomjegyzék


[AKS] **Agrawal, M., Kayal, N.,Saxena, N.:** "*Primes is in* P." Ann. Math. 160 (2004), 781-793., HTTP://WWW.CSE.IITK.AC.IN/USERS/MANINDRA/ALGEBRA/PRIMALITY_V6.PDF

[BD] **Bresoud, D.:** *Factorization and Primality Testing*, Springer Verlag, 1988.

[CLR] **Cormen, T., Leiserson, Ch., Rivest, R.:** *Algoritmusok*, Műszaki Kiadó 1999

[CsL] **Csete Lajos:** *Az Euklideszi algoritmus legrosszabb esete és Gabriel Lamé*, Polygon 1994, 83-88.

[CsL] **Csirmaz László:** *The Dealer's Random Bits inPerfect Secret Sharing Codes*, Studia Sci. Math. Hung. 32 (1996), pp.429-437.

[DT] **Dénes Tamás:** *Pierre Fermat és a nyilvános kulcsú rejtjelezés*, KöMaL 2001/8, 450-459.old.

[FR] **Freud Róbert:** *Ősi problémák – új eredmények (Számelmélet, prímszámok)*, HTTP://MATEK.FAZEKAS.HU/PORTAL/ELOADAS/2005/ELOADAS_2005_11_22_FREUD.HTML, 2008.

[FGy] **Freud Róbert, Gyarmati Edit:** *Számelmélet*, Nemzeti Tankönyvkiadó, Bp. 2006,

[IA] **Iványi Antal** (szerk.): *Informatikai algoritmusok*, HTTP://COMPALG.INF.ELTE.HU/~TONY/ELEKTRONIKUS/INFORMATIKAI/INFALG1E.PDF, 2007.

[JA] **Járai Antal:** *Számítógépes számelmélet*, HTTP://COMPALG.INF.ELTE.HU/~AJARAI/CNT.PDF, 1998.

[JI] **Járási István:** *A Carmichael-számokról*, KöMaL 2000/3, 136-145.old.

[KRSz] **Katona Gyula, Recski András – Szabó Csaba:** *A számítástudomány alapjai*, Typotex Kiadó, Budapest, 2002, ISBN 963-9326-24-0.

[KE1] **Kiss Elemér:** *Kétszáz éve született Bolyai János,* KöMaL 2002/november, 457-466.oldal

[KE2] **Kiss Elemér:** *Matematikai kincsek Bolyai János kéziratos hagyatékából*, Akadémiai Kiadó és Typotex Kft. Budapest, 1999., HTTP://MEK.OSZK.HU/05300/05321, HTTP://MEK.OSZK.HU/05300/05321/05321.PDF.

[KN] **Koblitz, N.:** *A Course in Number Theory and Cryptography*, Springer-Verlag, 1987.

[KD] **Knuth, D.:** *A számítógép-programozás művészete,* Műszaki Kiadó, 1988

[LL1] **Lovász László:** *An Algorithmic Theory of Numbers and Convexity*, SIAM, Philadelphia, 1986, ISBN 0-89871-203-3.

[LL2] **Lovász László:** *Algoritmusok bonyolultsága*, ELTE TTK jegyzet, Tankönyvkiadó Bp. 1990,
**bővített kiadás:** HTTP://WWW.CS.ELTE.HU/~KIRALY/ALG.PDF.

[LG] **Lovász László, Gács Péter:** *Algoritmusok*, Tankönyvkiadó, 1987.

[LM] **Laczkovich Miklós**: *Prímképletek,* KöMaL 47 (1997) 385-398.

[MGy] **Maróti György:** *Előadások algoritmikus számelméletből*, Livermore Kiadó, Békéscsaba, 2008.

[GC] **McGeoch, Catherine C.:** *Zero-Knowledge Proofs,* Computer Sci. Sampler, 1993, pp.682-685.

[MZ] **Megyesi Zoltán:** *Titkosírások*, Polygon IV/2, (1994).

[RISz] **Rónyai Lajos, Ivanyos Gábor, Szabó Réka:** *Algoritmusok,* Typotex Kiadó, 2005.

[RF] **Rozgonyi-Borus Ferenc:** *Számelméleti algoritmusok*,
HTTP://WWW.ABAX.HU/INLAP/T/CIKK/SZAMALG/SZAMALG.HTM, 2002.

[SA1] **Sárközy András:** *Számelmélet* („Bolyai könyvek” sorozat), Műszaki Kiadó, Bp. 1976.

[SA2] **Sárközy András:** *Számelmélet és alkalmazásai*, („Matematika Műszakiaknak” sorozat), Műszaki Kiadó, Bp. 1978.

[SzM] **Szalay Mihály:** *Számelmélet*, Tankönyvkiadó, Bp. 1991.
.

[SzI0] **Szalkai István:** *Újabb kombinatorikai feladatok,*
HTTP://MATH.UNI-PANNON.HU/~SZALKAI/KOMBUJ+FELAD2C.DOC

[SzI1] **Szalkai István:** *Algebra és számelmélet feladatgyűjtemény*, Pannon Egyetem Kiadó, Veszprém, 2005.

[SzI2] **Szalkai István:** *Diszkrét matematika és algoritmuselmélet alapjai*, Pannon Egyetem Kiadó Veszprém, 2001, javított kiadás: 2006.

[TV] **Tivolt Viktor:** *Prímszámtesztelő és -faktorizációs algoritmusok*, Szakdolgozat, Pannon Egyetem, 2005.

[W] HTTP://MATHWORLD.WOLFRAM.COM/

# Tárgymutató