Érdekességek a legnagyobb közös osztóról

- Tóth Géza
- informatika tanári rövid képzés

- 2020. március 6.

A legnagyobb közös osztó

- Két (nem egyszerre nulla) egész szám közös osztói közül a lehetséges legnagyobb pozitív egész, amely mindkét egész számot osztja.

- d közös osztója az a és b egész számoknak, ha d|a és d|b
- g a legnagyobb közös osztója a -nak és b -nek, ha g a közös osztók közül a legnagyobb.
- Jelölés: g = lnko(a;b) gcd(a;b) (greatest common divisor)

A legnagyobb közös osztó tulajdonságai

Az a és b számok bármely közös osztója az lnko(a, b)-nek is.

lnko(a, b) = lnko(b, a)

lnko$(a, a) = a$

c·lnko(a, b) = lnko$(c·a, c·b)$ (tetszőleges c számra)

lnko(a, b) = lnko$(a+bc, b)$

lnko$(a, b) = a$, akkor és csak akkor, ha $a|b$, azaz a osztója b-nek

ha lnko$(a, b) = 1$ és lnko$(a, c) = 1$, akkor lnko$(a, b·c) = 1$

ha $a|b·c$ és lnko$(a, b) = 1$, akkor $a|c$

lnko(m;n) kiszámolása

- Prímtényezős felbontás. Ehhez kell a prímszámok ismerete $\sqrt{n}$ -ig , ha n > m.

Ha elég nagy m és n, akkor ehhez nagyon nagy adatbázis kell, vagy nagyon sok számolás. 2016-ban a legnagyobb prímszám 22 millió számjegyű volt. Nem írtam fel.

Időigény:

lnko(m;n) kiszámolása prímtényezés felbontással: idő!

Egy n -jegyű N szám prímtényezős felbontását keressük. Módszer: a páratlan számokat próbáljuk ki √N -ig. Hány osztást kell elvégeznünk?

Ha az N szám n -jegyű, akkor értéke körülbelül $N \sim 10^n$ (pontosabban $10^{n-1} \leq N < 10^n$
Ekkor az osztások száma √N/2 .

5 GHz esetén:

n=20 esetén $5 \cdot 10^9$ lépés = 1 mp,
n=30 esetén $5 \cdot 10^{14}$ lépés = 10^5 mp ~ 27 óra 46 perc,
n=40 esetén $5 \cdot 10^{19}$ lépés = 10^{10} mp ~ 317 év 35 nap 18 óra,
n=50 esetén $5 \cdot 10^{24}$ lépés = 10^{15} mp ~ 31.7 millió év

lnko(m;n) kiszámolása prímtényezés felbontással: idő!

Mennyire csökkenne a futásidő, ha a √N alatti prímszámokat egy tömbben (táblázatban) tárolnánk, és csak e prímszámokat próbálnánk ki ?

Jelöljük tetszőleges x szám esetén π(x) -el az x -nél kisebb prímszámok számát!
A "Nagy Prímszámtétel" :
π(x) ~ x/ln(x).
Ekkor az osztások száma π(√N) ~ √N/ln(√N) .

n=20 (és 5GHz) esetén	~4.3·108 lépés < 1 mp,
n=30 esetén	~2.9·1013 lépés ~ 5790 mp ~ 1 óra 36 perc,
n=40 esetén	~2.2·1018 lépés ~ 4.4·108 mp ~ 13 év 281 nap 13 óra,
n=50 esetén	~1.4·1023 lépés ~ 1.7·1014 mp ~ 5.5 millió év

lnko(m;n) kiszámolása prímtényezés felbontással: idő!

- Láthattuk, hogy a „Nagy Prímszámtétel” felhasználásával a nagy számok prímtényezős felbontásának időigénye akár a hatodára is csökkenhet.
- Sajnos az évmilliónál nem sikerült kevesebb időt elérni.
- Az adattárolóból a kiolvasási idő is ront a helyzeten.
- Ha a prímszámokat a felbontás közben állítanánk elő, még rosszabbá válna a helyzetünk.

A „Nagy Prímszámtétel”

- Gauss (német, 1777 – 1855, 1796. május 30-án megsejtette, hogy

$$\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\ln(x)}} = 1$$

- Csebisev (orosz, 1821-1894) elemi módszerrel bebizonyította, hogy

$$0,922\frac{x}{\ln(x)} < \pi(x) < 1,105\frac{x}{\ln(x)}$$

- Hadamard és de La Vallée Poussin 1896-ban komplex függvénytani módszerekkel bebizonyította Gauss sejtését.

A „Nagy Prímszámtétel”

- Erdős Pál (1913 – 1996) 1949- ben Atle Selberggel elemi bizonyítást adott a prímszámtételre.
- Atle Selberg (norvég – USA, 1917 – 2007): 1986: Wolf-díj

Az euklideszi algoritmus

- Euklidesz i. e. 300 körül született, Platón akadémiáján tanult Athénban. Az alexandriai matematikai iskola megalapítója. Ezért ő Alexandriai Euklidesz. Az eljárást tetszőleges szakaszokra írta le, azaz tetszőleges pozitív valós számokra definiálta és határozta meg algoritmikusan a legnagyobb közös osztót.

Az euklideszi algoritmus

- Euklideszi algoritmus lnko(a;b) meghatározására:
- Adott $a;b \in Z$ 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

$a = b \cdot q_1 + r_1 , 0 < r_1 < |b|;$

$b = r_1 \cdot q_2 + r_2 , 0 < r_2 < |r_1|;$

$r_1 = r_2 \cdot q_3 + r_3 , 0 < r_3 < |r_2|;$

.

$r_{i-2} = r_{i-1} \cdot q_i + r_i ; 0 < r_i < |r_{i-1}|$

$r_{m-2} = r_{m-1} \cdot q_m + r_m ; 0 < r_m < |r_{m-1}|$

$r_{m-1} = r_m \cdot q_{m+1}$

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

Az euklideszi algoritmus „tesztelése”

- 1.) minden input esetében véges sok lépésben megáll, mert a maradékok csökkennek (véges egész számok!);
- 2.) az idő: legfeljebb a kisebb szám számjegyei számának 5-szöröse lépés után vége az algoritmusnak: $m = 5 \cdot \log_{10}|b|$ Gabriel Lamé (1795-1870) francia matematikus
- 3.) helyes eredményt ad, teljes indukcióval belátható;
- 4.) nehézsége, bonyolultsága: lásd 2. pont;

A legnagyobb közös osztó felhasználása

- prímtesztelésnél; (titkosítás);
- lineáris Diofantikus egyenletek
- relatív prímek ellenőrzésénél és keresésénél;
- törtműveleteknél (törtek egyszerűsítése);
- lineáris kongruenciáknál és kongruenciarendszereknél;
- lkkt(a;b) = a · b : lnko(a;b)
- …

A Fermat-prímteszt

Fermat (1607 – 1665, francia) Leibniz (1646 – 1716, német) bizonyította

- A kis Fermat-tétel: ha p prímszám és a < p egész szám, akkor $a^{p-1} \equiv 1 \pmod{p}$
- A teszt menete: Meg kívánjuk vizsgálni, hogy *n* szám *1*-nél nagyobb páratlan egész prím-e. Legyen $1<a<n$. Euklideszi algoritmussal ellenőrizhető, hogy *n* és *a* relatív prímek. Ha nem azok, akkor *n* összetett.
- Ha *n* prím, akkor a^{n-1} kongruens $1 \pmod{n}$. Ha nem így van, akkor *n* összetett.
- Ha igen, akkor újabb véletlen *a*-val folytatódik a vizsgálat.
- A legtöbb összetett szám ugyanis legfeljebb *1/2* valószínűséggel állja a tesztet egy véletlen *a*-ra.
- Futási ideje moduláris hatványozással $O(k \times \log^2 n \times \log\log n \times \log\log\log n)$, ahol *k* a fordulók száma, azaz ennyi véletlen alapra megy a tesztelés.
- A **Fermat-prímteszt** (vagy Fermat-féle prímszámpróba) egy valószínűségi prímteszt.

RSA titkosítás és az lnko

- Euler- féle $\phi(n)$ függvény: az n-nél kisebb, n-hez relatív prímek száma.

  (Leonhard Euler (1707-1783) svájci matematikus.)
- Speciálisan, ha p és q egymástól különböző prímek, akkor
- $\phi(p) = p-1$
- $\phi(p \cdot q) = (p-1) \cdot (q-1)$
- Ezeket a tételeket is felhasználva működik az RSA- titkosítás, és kell még egy

  $1 < e < \phi(p \cdot q)$ egész szám, amelyre lnko$(e; \phi(p \cdot q)) = 1$.

RSA- eljárás: Ron **R**ivest, Adi **S**hamir és Len **A**dleman fejlesztette ki 1976!!!-ban

Lineáris Diofantoszi egyenletek

- Legyenek adottak: a_1 ; a_2; … $a_{n;}$ c.
- Keresettek olyan x_1 ;x_2… x_n, amelyekre

- $a_1x_1 + a_2x_2 + … + a_nx_n = c$

- Bizonyítható, hogy ez az n változós lineáris Diofantoszi egyenlet pontosan akkor oldható meg, ha
- lnko(a_1; a_2… a_n) | c

$u_0 \cdot a + v_0 \cdot b$

- Minden a és b egész számhoz léteznek olyan u_0 és v_0 egész számok, melyekre lnko(a;b) = $u_0 \cdot a + v_0 \cdot b$
- Példa: lnko(9867; 8855) = $9 \cdot 9867 + (-10) \cdot 8855 = 88803 - 88550 = 253$
- $u_0 \cdot= 9;$ $v_0 = -10$
- Hogyan találtuk meg ezeket a számokat?

u_0 · és v_0 keresése, kiterjesztett Euklideszi algoritmus

- $9867 = 8855 \cdot 1 + 1012$
- $8855 = 1012 \cdot 8 + 759$
- $1012 = 759 \cdot 1 + 253$
- $759 = 253 \cdot 3 + 0$

Alulról felfelé:

$$253 = 1 \cdot 1012 + (-1) \cdot 759 =$$
$$1 \cdot 1012 + (-1) \cdot (8855 - 8 \cdot 1012) =$$
$$1 \cdot 1012 + (-1) \cdot 8855 + 8 \cdot 1012 =$$
$$9 \cdot 1012 + (-1) \cdot 8855 =$$
$$9 \cdot (9867 - 8855) + (-1) \cdot 8855 =$$
$$9 \cdot 9867 + (-10) \cdot 8855$$

$u_0 = 9$; $v_0 = -10$

9867x + 8855y = 759 egy megoldása

- Van-e megoldás?
- Pontosan akkor van megoldás, ha lnko(a_1; a_2… a_n) | c , azaz
- lnko(9867; 8855) | 759
- lnko(9867; 8855) = 253 és 759 = 253 · 3
- Van megoldás.

9867x + 8855y = 759 egy megoldása

- A korábbi számításokat felhasználva:
- 9867x + 8855y = 759
- 9867x + 8855 x = 3 · 253
- 9867x + 8855y = 3 · (9 · 9867 – 10 · 8855)
- 9867x + 8855y = 27 · 9867 – 30 · 8855
- azaz x = 27; y = -30
- Vegyük észre, hogy $x = 3 \cdot u_0$ és $y = 3 \cdot v_{0,}$ ahol 3 = 759 : lnko(9867; 8855)
- Vegyük észre, hogy meg kellett jegyeznünk az összes maradékot, és kétszer kellett számolnunk!

9867x + 8855y = 759 összes megoldása

- Bizonyítható, hogy ha fenti egyenletnek van egy x_0; y_0 megoldása, akkor az összes gyök:
- $x = x_0 + k \cdot \frac{b}{lnko(a;b)}$ $\quad y = y_0 - k \cdot \frac{a}{lnko(a;b)}$

- Azaz például k = 1 illetve k = -2 esetén

$x = 27 + 1 \cdot \frac{8855}{lnko(9867;8855)} = 62$ $\quad y = -30 - 1 \cdot \frac{9867}{lnko(9867;8855)} = -69$

$x = 27 + (-2) \cdot \frac{8855}{lnko(9867;8855)} = -43$ $\quad y = -30 - (-2) \cdot \frac{9867}{lnko(9867;8855)} = 48$

is megoldás.

A kiterjesztett Euklideszi algoritmus optimalizálása

Végezzük el ismét az euklideszi algoritmust a lnko(9867; 8855) megkeresésére az oldalsó számolással együtt!

- $9867 = 8855 \cdot \mathbf{1} + 1012$
- $8855 = 1012 \cdot \mathbf{8} + 759$
- $1012 = 759 \cdot \mathbf{1} + 253$
- $759 = 253 \cdot 3 + 0$

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

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

A kiterjesztett Euklideszi algoritmus optimalizálása

- lnko(9867; 8855) = 253 $\xi_3 = 1 - (-8) \cdot 1 = 9;$ $\zeta_3 = -1 - 9 \cdot 1 = -10,$

azaz

- ξ_3 és ζ_3 egy megoldása a $9867\xi + 8855\zeta = 253$ egyenletnek.
- $\xi_3 = u_0$; $\zeta_3 = v_0$
- $X_0 = \xi_3 \cdot \frac{759}{253} = 27$ $y_0 = \zeta_3 \cdot \frac{759}{253} = -30$

A kiterjesztett Euklideszi algoritmus optimalizálása

- Vegyük észre, hogy az algoritmust csak egyszer futtattuk;
- Nem kellett erőforrást áldoznunk a maradékok tárolására!

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = c$$

- Ha megoldás létezik, akkor az előáll a bemutatott eljárás többszöri alkalmazásával. Konkrét leírását az
- ax + by + cz = m diofantoszi egyenletnek találtam.
- Sajnos az idő rövidsége és az én megfontolt haladásom nem teszi lehetővé ennek bemutatását.
- Ugyanígy jártam a Kínai maradéktétellel, amelynek megoldásához ismernünk kell a modulusok lkkt-ét.
- A nagy számok szorzása szintén kongruencia rendszer megoldására vezethető vissza.

Felhasznált irodalom:

dr. Szalkai István, dr. Dósa György: Algoritmikus számelmélet
Pannon Egyetem, Veszprém, 2011
https://math.uni-pannon.hu/~szalkai/KombUj+Felad2c.pdf
https://hu.wikipedia.org/wiki/Carl_Friedrich_Gauss
https://hu.wikipedia.org/wiki/Pafnutyij_Lvovics_Csebisev
https://hu.wikipedia.org/wiki/Pr%C3%ADmsz%C3%A1mt%C3%A9tel
https://hu.wikipedia.org/wiki/Erd%C5%91s_P%C3%A1l#Sz%C3%A1melm%C3%A9let
https://math.uni-pannon.hu/~szalkai/Euklideszi_Inko.pdf
https://hu.wikipedia.org/wiki/Kis_Fermat-t%C3%A9tel

Köszönöm a figyelmet!