A kis Fermat-tétel gyakorlati alkalmazása

Seregi Benjámin Martin

2009. október 29.

> "[...] Ezért a bölcs sürgés nélkül működik, szó nélkül tanít, nézi az áramlást és hagyja, nem erőlködik, alkot, de művét nem birtokolja, cselekszik, de nem ragaszkodik, beteljesült művét nem félti, s mert magának nem őrzi, el se veszíti." Lao-ce

1. Bevezetés

A kis Fermat-tétel egy számelméleti tétel, amely az egyik legalapvetőbb kongruenciatétel (egész számok maradékaival foglalkozó számelméleti tudományterület), ezért szokás nevezni Fermat kongruenciatételének is.

A kis Fermat-tétel elnevezés annak tudható be, hogy Pierre de Fermat (1601-1665) matematikus nevéhez fűződött egy másik fontos tétel is, amelyet nagy Fermat-tételnek neveztek és bizonyítása több száz évig húzódott el.


Pierre de Fermat

Fermat Frenicle de Bessynek írt levelében azt állította, hogy ha p prímszám, akkor osztója $(a^p - a)$-nak, bizonyítani azonban nem tudta, ahogyan másik híres tételét a nagy Fermat-tételt sem. Az első helyes bizonyítást, az akkor még csak sejtésnek nevezhető állításra Gottfried Wilhelm Liebniz adta.

Bár a nagy Fermat-tétel ($x^n + y^n = z^n$ egyenletnek nincs megoldása, ha $x, y, z \in \mathbb{Z} \setminus \{0\}$ és $n \in \mathbb{N}, n \geqq 3$) bizonyítása egy sokkal nehezebb matematikai probléma volt, mégis a gyakorlati alkalmazásokban a kis Fermat-tételnek jut a nagyobb szerep.

A modern kriptográfiában az egyik legszélesebb körben elterjedt titkosítási eljárás az RSA algoritmus, amelynek működése a nagy (1024 bites vagy annál nagyobb) véletlen prímszámokra és azok szorzatának nehéz faktorizációjára[1] épül.

Nagy véletlen prímszámokat determinisztikus módszerekkel (pl.: osztók végigpróbálgatása) keresni aszimptotikusan rossz (pl.:$\Theta(\sqrt{n})$) komplexitású[2] meg-

[1] Összetett számok prímtényezőkre bontása.
[2] Az algoritmusok hatékonysága (futásideje, memóriahasználata).

oldásokat eredményeznek, ezért a gyakorlati alkalmazásokban nem használhatóak eredményesen.

A kis Fermat-tétel adja a gyakorlatban leggyakrabban használt Miller-Rabin prímtesztelő eljárás matematikai alapjait, ami szerves részét képezi az RSA algoritmusnak.

2. A kis Fermat-tétel bizonyítása

A kis Fermat tétel definíciója[3] a kongruenciaelmélet segítségével a következőképpen néz ki, ha p prímszám és $a \in \mathbb{Z}$, akkor:

$$a^p \equiv a \,(mod\, p)$$

Bizonyítás. A kis Fermat-tétel bizonyításához szükséges egy segédtétel, úgynevezett lemma, ami a binomiális együtthatók tulajdonságáról szól.

Lemma. *Minden $(x+y)^p$ binomiális együttható (kivéve az első és az utolsó tagot) osztható p-vel, ha p prímszám.*

Tudjuk, hogy a binomiális tétel általános formája a következő:

$$(x+y)^p = \sum_{k=0}^{p} \binom{p}{k} x^{p-k} y^k = \binom{p}{0} x^p y^0 + \binom{p}{1} x^{p-1} y^1 + ... + \binom{p}{p} x^0 y^p$$

Amennyiben azt akarjuk megmutatni, hogy $p \,|\, (x+y)^p - x^p - y^p$, akkor $\binom{p}{k}$ kifejezésre kell koncentrálni, hiszen az minden tagnak szorzótényezője. Könnyű belátni, hogy $\binom{p}{k} \equiv 0 \,(mod\, p)$, ha p prímszám, és $0 < k < p$, mert $\binom{p}{k}$ felbontása a következő:

$$\binom{p}{k} = \frac{p!}{k!(p-k)!} = \frac{p \cdot (p-1) \cdot ... \cdot (p-k+1)}{k!}$$

Azt mondhatjuk tehát, hogy $\binom{p}{k}$ tartalmazni fogja p tényezőt, ha p prím és $0 < k < p$, hiszen $k!$ soha nem lesz osztható p-vel. Ez a prímek tulajdonságából következik, ha $k!$ osztható lenne p-vel, akkor p nem lenne prím, hiszen tartalmazna önmagánál kisebb prímtényezőket és ez ellentmondás lenne (ehhez kapcsolódik egy másik fontos tétel, a Wilson-tétel).

Láthatjuk, hogy a binomiális tétel összegzési képletében csupán kétszer nem áll fent $0 < k < p$ egyenlőtlenség, az első és az utolsó tagban, ezért ha eltávolítjuk ezt a két tagot, akkor az összeg osztható lesz p-vel. A lemma általánosítása a következőképpen néz ki:

$$(x+y)^p \equiv x^p + y^p \,(mod\, p)$$

[3] A következő alakban is előfordul: $a^{p-1} \equiv 1 \,(mod\, p)$.

■

A kis Fermat-tétel bizonyításához ennek a lemmának egy speciális esetét fogjuk alkalmazni.

Teljes indukcióval bizonyítható az állításunk, $a = 1$-re a tétel állítása nyilvánvalóan teljesül, hiszen 1 bármely hatványa szintén 1, tegyük fel, hogy az állításunk igaz $a + 1$-re, ekkor a következőt kapjuk:

$$(a+1)^p = a^p + \binom{p}{1} a^{p-1} + ... + 1$$

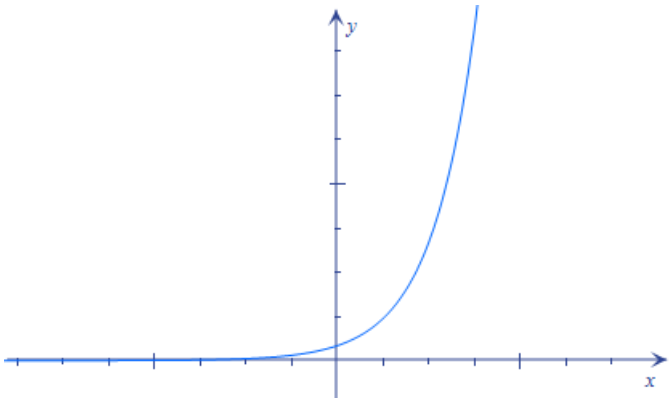
Könnyen belátható, hogy $(a + 1)^p$ p-vel vett maradéka 1-el nagyobb, mint a^p p-vel vett maradéka, ezért megegyezik $(a + 1)$ maradékával.

■

3. A moduláris hatványozás

Amennyiben a kis Fermat-tételt a gyakorlatban is alkalmazni akarjuk, mint prímteszt, felmerül a kérdés, hogyan kezeljük az a^p számokat, ha p nagy szám. Igazából p-nek nem is kell túl nagynak lennie, hiszen az exponenciális növekedésnek köszönhetően a hatványérték már meglehetősen kis p-re is nagy lesz. A nagy számokkal két probléma van, az egyik, hogy nehéz tárolni őket az alapvető adattípusokban (32 és 64 bites egészek), valamint az, hogy a rajtuk végzett aritmetikai műveletek hossza a szám bitjeinek függvényében növekszik.

Ezért volt szükség egy algoritmusra, amivel gyorsan meghatározható $a^p \, mod \, p$ értéke, a^p tényleges kiszámítása nélkül.



Az exponenciális növekedés

Ahhoz, hogy a moduláris hatványozást megérthessük, először a gyorshatványozás fogalmával kell megismerkedni, vegyük a következő példát: számoljuk ki a^b értékét gyorsan, ha $a = 3, b = 13$.

Legyen $\langle b_k, b_{k-1}, ...b_0 \rangle$ b bináris reprezentációja (ahol k a legnagyobb helyiérték), ekkor b felírható a 2 hatványainak összegeként $b = b_k 2^k + b_{k-1} 2^{k-1} + ... + b_0 2^0$, tehát a példához visszatérve:

$$3^1 \cdot 3^4 \cdot 3^8 = 3^{13}$$

Ez a hatványozás alapazonosságából adódik ($a^k \cdot a^l = a^{k+l}$).

A következő táblázatban jól látszik, hogy a legkisebb helyiértéktől indulunk, és úgy végezzük el az ismételt négyzetre emelést, csupán egy 0-ás bit van a 13-ban, ezért a 3^2-nek nem kell szerepelnie a szorzatban (a táblázatban csupán a szemléltetés miatt van elosztva a produktum 9-el, az algoritmus valójában nem szoroz, ha 0-ás bithez érkezik, de ez a pszeudokód 4. sorában definiálva is van).

3^1	3^2	3^4	3^8	$\prod_9 = 3^{13}$
1	0	1	1	$(1101)_2 = 13$
3	9	81	6561	1594323

Algoritmus 1 Gyorshatványozás algoritmusa

Input: $a, b \rightarrow a^b$
1. $e \leftarrow 1$
2. $k \leftarrow 0$
3. *Amíg* $k <> \lfloor log_2(b) \rfloor$
4. *Ha* b_k *igaz*
5. $e \leftarrow a \cdot e$
6. $k \leftarrow k + 1$
7. $a \leftarrow a \cdot a$
8. *kinyomtat* e

Jól látható, hogy ennek az algoritmusnak b bináris hosszától függ a lépésszáma, vagyis a komplexitása $\Theta(\lfloor log_2 b \rfloor)$. A gyorshatványozó algoritmus a hatványok azon tulajdonságát használja ki, hogy:

$$a^{2c} = (a^c)^2$$

$$a^{2c+1} = a \cdot (a^c)^2$$

Ugyanezek az állítások igazak, akkor is, ha a maradékokat nézzük, hiszen a jobb és bal oldal ekvivalens, ezért ekvivalens maradékot is kell adniuk:

$$a^{2c} \, mod \, n = (a^c)^2 \, mod \, n$$

$$a^{2c+1} \, mod \, n = a \cdot (a^c)^2 \, mod \, n$$

Ennek következményeként a gyorshatványozó algoritmus könnyen átalakítható moduláris hatványozó algoritmussá:

Algoritmus 2 Moduláris hatványozás algoritmusa

Input: $a, b, n \rightarrow a^b \, mod \, n$
1. $e \leftarrow 1$
2. *Amíg* $k <> 0$
3. $\quad$ *Ha* b_k *igaz*
4. $\quad\quad e \leftarrow (a \cdot e) \, mod \, n$
5. $\quad k \leftarrow k - 1$
6. $\quad a \leftarrow (a \cdot a) \, mod \, n$
7. *kinyomtat* e

Most már a kezünkben van minden eszköz, ami szükséges ahhoz, hogy hatékony prímtesztelő eljárást írhassunk.

4. A Fermat-féle prímteszt

A Fermat-féle prímteszt egy valószínűségi prímteszt, ez azt jelenti, hogy egy tetszőleges p számról csak bizonyos valószínűséggel tudja megmondani, hogy összetett vagy prím, ez a tulajdonság annak köszönhető, hogy vannak olyan p számok, amelyek bizonyos a alapra álprímek, ez azt jelenti, hogy p ugyan összetett, de mégis teljesül rá a kis Fermat-tétel kongruenciája (emiatt nem igaz a kis Fermat-tétel megfordítása[4]).

Ezen számok kiszűrése nem okoz túl nagy problémát, hiszen több véletlen a alapra tesztelve a kongruenciát könnyedén kiszűrhetőek, azonban léteznek olyan univerzális álprímek, amelyek az összes a alapra álprímek, ezeket a számokat nevezzük Carmichael-számoknak. A Carmichael-számok meglehetősen ritkák, 10^8-ig csupán 255 darab létezik.

Ahhoz, hogy a Fermat-féle prímtesztet eredményesen tudjuk használni, fel kell ismerni a Carcmichael-számokat, valamint megfelelő valószínűségi korlát alá kell csökkenteni az álprímek teszten való átjutását.

Algoritmus 3 Fermat-féle prímteszt

Input: n, s
1. *Amíg* $s <> 0$
2. $\quad a \leftarrow$VÉLETLEN-SZÁM $(1, n-1)$
3. $\quad$ *Ha* MODULÁRIS-HATVÁNYOZÓ$(a, n, n) <> a$
4. $\quad\quad$ *kinyomtat* ÖSSZETETT
5. $\quad s \leftarrow s - 1$
6. *kinyomtat* VALÓSZÍNŰLEG-PRÍM

[4] Bár Bolyai János megpróbálta bebizonyítani, de rájött, hogy lehetetlen.

Láthatjuk, hogy az algoritmus egy tetszőleges n egészt és egy s, úgynevezett „biztonsági” változót vár, ami azt határozza meg, hogy az algoritmus hány véletlen a alapra tesztelje le a kongruenciát, ezzel a módszerrel az álprímek nagyon jól kiszűrhetőek, hiszen annak a valószínűsége, hogy egy 50 bites véletlenszám 2-es alapú álprím legyen, kisebb, mint egy a millióhoz.

4.1. A Carcmichael-számok

Az előzőekben kiderült, hogy a Carmichael-számok olyan univerzális álprímek, amelyek bármilyen a bázishoz álprímek, ezért a Fermat-féle prímteszt nem képes őket kiszűrni. Ezen számoknak két fontos tulajdonsága van (Korselt kritériumai):

561	1105	1729	2465
$3 \cdot 11 \cdot 17$	$5 \cdot 13 \cdot 17$	$7 \cdot 13 \cdot 19$	$5 \cdot 17 \cdot 29$

Az első négy Carmichael-szám és faktorizációjuk

Egy pozitív összetett n egész, akkor és csakis akkor Carmichael-szám, ha:

- n négyzetmentes (olyan szám, amely nem osztható 1-nél nagyobb természetes szám négyzetével)
- n minden p prímtényezőjére igaz, hogy $p - 1 \mid n - 1$.

Korselt kritériumainak bizonyítása megtalálható [1]-ben, a Fermat-féle prímteszt kiegészítését, amely nagy valószínűséggel képes kiszűrni a Carmichael számokat, Miller-Rabin-féle sztochasztikus prímtesztnek nevezzük, ennek az algoritmusnak létezik egy determinisztikus variánsa is, azonban annak komplexitása nem polinom[5], így a gyakorlatban nem alkalmazzák.

Az első polinomiális futásidejű determinisztikus prímtesztelő algoritmust (AKS-algoritmus), csak hét évvel ezelőtt sikerült megalkotni és matematikailag bizonyítani.

Irodalomjegyzék

- [1] Borwein, D.; Borwein, J. M.; Borwein, P. B.; and Girgensohn, R. "Giuga's Conjecture on Primality." Amer. Math. Monthly 103, 40-50, 1996.
- [2] Crandall, R. and Pomerance, C. Prime Numbers. New York: Springer-Verlag, 2001.
- [3] Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. Introduction to Algorithms. Cambridge, MA: MIT Press, 1990.

[5] Puritán módon megfogalmazva: gyors, hatékony.

- [4] Ore, Ø. Invitation to Number Theory. Washington, DC: Math. Assoc. Amer., 1967.
- [5] Agrawal, M.; Kayal, N.; and Saxena, N. "Primes is in P." Ann. Math. 160, 781-793, 2004.
- [6] M. O. Rabin. Probabilistic algorithm for testing primality. J. Number Theory, 12:128–138, 1980.
- [7] Carmichael, R. D.: On composite numbers P which satisfy the Fermat congruence $a^{p-1} \equiv 1 \, (mod \, p)$. Amer. Math. Monthly, 19(1912), 22–27.
- [8] A. Korselt: Problème chinois, L'Intermédiaire des Mathématiciens 6 (1899), 142-143.
- [9] Koshy, T. Elementary Number Theory with Applications. San Diego, CA: Harcourt/Academic Press, p. 121, 2001.
- [10] Higgins, P. M. Mathematics for the Curious. Oxford, England: Oxford University Press, 1998.