Testovanie prvočíselnosti II.

Kubo Kováč

19. apríla 2010

1 Zlepšujeme Fermata

V predchádzajúcej časti sme ukázali jednoduchý test založený na Fermatovej vete, ktorý mal však malý „bug“: nefungoval pre Carmichaelove čísla. V tejto časti to napravíme a Fermatov test vylepšíme. Pri odlišovaní zložených čísiel a prvočísiel nám pomôže ich ďalšia vlastnosť: Rovnica

$$x^2 \equiv 1 \pmod{n}$$

má iba 2 riešenia (mod n), ak n je prvočíslo (konkrétne ± 1). Ale ak n je zložené číslo, tak táto rovnica má *viac* riešení (výnimkou je číslo 4, ale ďalej sa zaoberajme iba nepárnymi číslami). Vieme dokonca, že pre nepárne n je tých riešení práve 2^k, kde k je počet rôznych prvočíselných deliteľov n.

Napríklad pre $15 = 3 \times 5$ má rovnica $x^2 \equiv 1 \pmod{15}$ štyri riešenia (mod 15): $1, 4, 11, 14$ (presvedčte sa o tom). Mimochodom, $14 = 15 - 1$ a $11 = 15 - 4$, teda $14 \equiv -1$ a $11 \equiv -4 \pmod{15}$ a tie štyri riešenia sa dajú napísať aj ako ± 1, ± 4.

Prečo je to tak? Ukážme si aspoň náznak dôkazu: Povedať, že $x^2 \equiv 1 \pmod{n}$ je to isté, ako povedať, že n delí $x^2 - 1$. Avšak, ako vieme, $x^2 - 1 = (x-1)(x+1)$, teda n delí $(x-1)(x+1)$. Ak n je prvočíslo väčšie ako 2, tak nemôže naraz deliť $x - 1$ aj $x + 1$. Teda n musí deliť buď $x - 1$ alebo $x + 1$. V prvom prípade dostaneme riešenie 1, v druhom $n - 1 \equiv -1$.

Čo sa však stane, ak je číslo n zložené? Dajme tomu, že $n = pq$ je súčinom dvoch prvočísiel (väčších ako 2). Situácia $pq \setminus (x-1)(x+1)$ však teraz môže nastať až štyrmi spôsobmi:

1. $pq \setminus x - 1$
2. $pq \setminus x + 1$,
3. $p \setminus x - 1$ a $q \setminus x + 1$, alebo
4. $q \setminus x - 1$ a $p \setminus x + 1$.

V prvých dvoch prípadoch dostaneme riešenia ± 1; to že prípady 3 a 4 môžu naozaj nastať tvrdí Čínska zvyšková veta (ktorej dôkaz tu neuvádzame). V predchádzajúcom príklade $n = 15 = 3 \times 5$ pre riešenia 4 a 11 naozaj platí:

$3 \setminus 4-1$, $5 \setminus 4+1$ a $5 \setminus 11-1$, $3 \setminus 11+1$. Vo všeobecnosti, ak je n nepárne a $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ je rozklad n na súčin prvočísiel, tak niektoré $p_i^{e_i}$ budú deliť $x-1$ a tie zvyšné budú deliť $x+1$ – počet týchto možností je práve 2^k a to, že naozaj môžu nastať zaručí Čínska zvyšková veta.

Ako nám táto vlastnosť pomôže pri rozlišovaní zložených čísiel a prvočísiel? Nuž akonáhle nájdeme také číslo (rôzne od ± 1), ktorého druhá mocnina je 1, vieme, že číslo *nemôže* byť prvočíslo. Takáto situácia môže nastať pri Fermatovom teste: tam počítame hodnotu $a^{n-1} \bmod n$ a ak vyjde 1, hovoríme, že n prešlo Fermatovym testom. Čomu sa ale potom rovná $a^{(n-1)/2} \bmod n$ (teda a umocnené na polovičný exponent)? Ak n je prvočíslo, tak to môže byť iba ± 1. V opačnom prípade sme našli číslo (rôzne od ± 1), ktoré umocnené na druhú dáva jednotku:

$$(a^{(n-1)/2})^2 = a^{n-1} \equiv 1 \pmod{n}.$$

Trochu všeobecnejšie: vo Fermatovom teste počítame nejaké hodnoty $a^k \pmod{n}$. Ak vieme, že $a^k \equiv 1 \pmod{n}$ a k je párne číslo, vypočítajme $a^{k/2} \bmod n$. Ak vyjde iné číslo ako ± 1, vieme, že n je zložené (v tomto prípade vieme dokonca nájsť netriviálneho deliteľa n). Tento test si nazvime „odmocninový test“[1]:

> Ak $a^k \equiv 1 \pmod{n}$, k je párne a
>
> $$a^{k/2} \not\equiv \pm 1 \pmod{n},$$
>
> tak číslo n je zložené.

Napríklad sme si povedali, že $3^{90} \equiv 1 \pmod{91}$; teda 91 prejde Fermatovym testom pre $a = 3$. Ak by sme však použili ešte odmocninový test a vypočítali $3^{45} \bmod 91 \equiv 27$, zistili by sme, že 91 je zložené číslo a teda 3 je klamár. Číslo 27 totiž umocnené na druhú dáva 1:

$$27^2 \equiv (3^{45})^2 = 3^{90} \equiv 1 \pmod{91}.$$

Iný príklad: povedali sme si, že najmenšie číslo, pri ktorom zaklame dvojka je 341. Naozaj $2^{340} \bmod 341 = 1$. Použime odmocninový test. Čo je $2^{340/2} = 2^{170} \pmod{341}$? Ukazuje sa, že výsledok je 1 – tá dvojka teda stále zatĺka. Avšak keďže sme dostali *jednotku* a 170 je *párne* číslo, môžeme opäť použiť odmocninový test a pýtať sa ďalej: Koľko je $2^{170/2}$? Zistíme, že $2^{85} \equiv 32 \pmod{341}$, čo je náš kýžený dôkaz, že 341 nemôže byť prvočíslo:

$$32^2 \equiv (2^{85})^2 = 2^{170} \equiv 1 \pmod{341}.$$

(Mimochodom, $341 = 11 \times 31$ a vidíme, že $31 \setminus 32-1$ a $11 \setminus 32+1$.) Všimnite si, že keby sme dostali -1, alebo namiesto 170 by sme mali nepárne číslo, nemohli by sme takto pokračovať.

Posledný príklad: Carmichaelovo číslo 561 a zoberme $a = 13$. Keďže 13 a 561 sú nesúdeliteľné, Fermatovym testom 561 prejde: $13^{560} \equiv 1 \pmod{561}$.

[1] toto nie je oficiálny ani zaužívaný názov

Číslo 560 je však párne a môžeme sa pýtať, koľko je 13^{280}. Odpoveď: $13^{280} \equiv 1 \pmod{561}$ (13 zatĺka). Keďže sme však dostali 1 a 280 je párne číslo, môžeme pokračovať vo „výsluchu“: koľko je 13^{140}? 13^{140} je stále 1 (mod 561), ale 140 je opäť párne číslo, pokračujeme: koľko je 13^{70}? A tu sa 13 „zlomí“ a prizná sa: $13^{70} \equiv 67 \pmod{561}$. Našli sme teda dôkaz, že 561 je zložená: $67^2 \equiv 1 \pmod{561}$.

Dostávame takýto vylepšený test, prezývaný *Millerov test*:

0. Zvolíme nejaké $1 < a < n$.
1. Spravíme Fermatov test: $a^{n-1} \bmod n$
2. Ak je výsledok rôzny od 1, n je zložené – neprešlo Fermatovym testom.
3. Kým máme kongruenciu tvaru $a^k \equiv 1 \pmod{n}$, kde k je párne, robíme odmocninový test – vypočítame $a^{k/2} \bmod n$. Ak je výsledok rôzny od ± 1, n je zložené – neprešlo odmocninovým testom.

Týmto testom môže prejsť iba prvočíslo, alebo „veľmi dobrý klamár“ – takéhoto klamára voláme *silné pseudoprvočíslo* (pri báze a).

Ako môže číslo n prejsť Millerovým testom? Musí prejsť Fermatovým testom a potom (ako delíme exponent) musíme dostávať samé jednotky, až kým k nie je nepárne, alebo niekedy dostať -1.

Samozrejme, v skutočnosti nepočítame a^{n-1} a potom odznovu $a^{(n-1)/2}$ a odznovu $a^{(n-1)/4}$, atď. Napíšme si $n-1$ ako

$$\underbrace{2 \times 2 \times \cdots \times 2}_{s} \times t,$$

t.j. $n - 1 = 2^s \cdot t$, kde t je už *nepárne*. Potom umocniť na $n-1$ znamená umocniť na t a potom ešte s-krát na druhú. Pri umocňovaní sa budeme pozerať, či nenájdeme číslo, ktorého druhá mocnina je 1.

Ukážme si to na príklade čísla 1409 (toto číslo je prvočíslo); platí: $1408 = 2^7 \times 11$, teda umocniť na 1409 znamená umocniť na 11 a potom sedemkrát na druhú. V nasledujúcej tabuľke sme spravili Millerov test pre prvočíselné a od 2 po 11 a potom ešte pre niektoré vybraté áčka.

a	a^{11}	$a^{2\times 11}$	$a^{2^2\times 11}$	$a^{2^3\times 11}$	$a^{2^4\times 11}$	$a^{2^5\times 11}$	$a^{2^6\times 11}$	$a^{2^7\times 11} = a^{1408}$
2	639	1120	390	1337	957	1408	1	1
3	1022	4150	327	1254	72	957	1408	1
5	639	1120	390	1337	957	1408	1	1
7	957	1408	1	1	1	1	1	1
11	185	409	1019	1337	957	1408	1	1
49	1408	1	1	1	1	1	1	1
85	112	1272	452	1408	1	1	1	1
115	1	1	1	1	1	1	1	1

Všimnime si, že $1^2 = 1$, to znamená, že ak raz dostaneme jednotku, zvyšok riadku budú samé jednotky. Tiež si uvedomme, že $-1 \equiv n - 1 \pmod{n}$, teda to, čomu hovoríme -1 je v našom príklade to isté ako 1408.

Ak si vždy nakreslíme tabuľku ako v predchádzajúcom príklade, môžeme oba testy formulovať aj takto:

1. Číslo prejde Fermatovým testom, ak je v poslednom stĺpci jednotka.
2. Číslo prejde odmocninovým testom, ak sú v celom riadku iba jednotky (napríklad $a = 115$ v predchádzajúcom príklade), alebo je pred prvou jednotkou -1, resp. $n - 1$ (v predchádzajúcom príklade 1408).

Implementácia je teraz jednoduchá: Majme dané n, s, t, a, kde $n-1 = 2^s \times t$, pričom t je nepárne. Najskôr vypočítame a^t; ak je to jednotka, môžeme skončiť – číslo n prešlo testom. V opačnom prípade budeme počítať mocniny

$$a^t, a^{2t}, a^{2^2 t}, \ldots, a^{2^{s-1} t}.$$

Ak je aspoň jedno z týchto čísiel $n - 1$, môžeme skončiť, pretože ďalšie číslo bude 1 a tiež všetky ostatné; n prešlo testom. Naopak, ak žiadne z nich nie je $n - 1$, tak podľa toho, čo je $a^{2^s t} = a^{n-1}$, n buď neprejde Fermatovým, alebo odmocninovým testom – číslo n je zložené.

Algoritmus 1 Implementácia Millerovho testu (vypočítame $a^t \bmod n$, potom umocňujeme na druhú a pozeráme sa, čo vychádza). Predpokladáme tu, že $n - 1 = 2^s \times t$; hodnoty s a t sa z n dajú vypočítať veľmi jednoducho, ako ukazuje program vpravo.

```
from random import randint

def strong (n, s, t, a):
   x = modexp(a, t, n)
   if x == 1: return True
   for i in xrange(0,s):
      if x == -1: return True
      x = (x*x) % n
   return False
```

```
def get_st (n):
   s, t = 0, n-1
   while t%2 == 0:
      s = s + 1
      t = t / 2
   return s, t
```

Ak vieme, ako test implementovať, poďme si povedať o jeho úspešnosti/neúspešnosti v rozpoznávaní prvočísiel. Keďže tento test v sebe už zahŕňa aj Fermatov test, Millerov test nebude horší. Otázne je, či je nejako podstatne lepší. Nakoľko nám naše vylepšenie pomôže? Stačí teraz dané číslo otestovať pomocou niekoľkých a-čok? Stačí nám jediné?

☹ *Zlá správa #1.* Vykonať Millerov test iba pre $a = 2$ nám nezaručí správny výsledok: existuje nekonečne veľa zložených čísiel, ktoré sa „tvária“ ako prvočísla a prejdú týmto testom pre $a = 2$. Niektoré takéto vieme aj povedať: Ak $p > 5$ je

prvočíslo, tak $n = (4^p + 1)/5$ je zložené číslo, ktoré prejde Millerovým testom, teda je silné pseudoprvočíslo pri báze 2.

☹ *Zlá správa #2.* Existuje dokonca nekonečne veľa silných pseudoprvočísiel pre ľubovoľnú bázu. T.j. nech si zoberieme hocijaké konkrétne a, pre každé existuje nekonečný zástup klamárov, ktoré testom prejdú.

☹ *Zlá správa #3.* Matematici dokonca dokázali, že ak pre zložené n označíme $W(n)$ veľkosť toho najmenšieho svedka, ktorý dokazuje, že n je zložené, potom pre nekonečne veľa n platí:

$$W(n) > (\ln n)^{1/(3 \ln \ln \ln n)}.$$

Inými slovami: keby sme zloženosť čísla n testovali tak, že vyskúšame postupne $a = 2, 3, 4, \ldots$ až kým nenájdeme nejaké a, ktoré dosvedčí, že n je zložené (alebo nevyskúšame všetky), tak pre nekonečne veľa zložených čísiel musíme vyskúšať aspoň $(\ln n)^{1/(3 \ln \ln \ln n)}$ a-čiek, kým nenájdeme svedka.

Keďže hodnota $(\ln n)^{1/(3 \ln \ln \ln n)}$ s rastúcim n rastie donekonečna, vidíme, že nestačí ani len nejaká konkrétna sada zopár a-čiek. Nestačí otestovať číslo n pre $a = 2$, 3 a 47 a podľa toho, či prejde všetkými testami ho vyhlásiť za prvočíslo alebo zložené číslo. Pre ľubovoľne veľkú konečnú sadu a-čiek budú existovať prefíkaní klamári, ktorí testom prejdú.

To boli zlé správy. Na druhej strane však máme veľa dobrých správ:

☺ *Dobrá správa #1.* Ak je pravdivá tzv. rozšírená Riemannova hypotéza (asi najdôležitejšie *nedokázané* tvrdenie v matematike), stačí test spustiť pre všetky $a < 2 \ln^2 n$. Ak nájdeme svedka, číslo je zložené; ak nie, môžeme číslo n smelo (ak veríme rozšírenej Riemennovej hypotéze) prehlásiť za prvočíslo.

Ak teda táto hypotéza platí, máme pred sebou *polynomiálny* algoritmus! T.j. taký, ktorého čas sa dá zhora ohraničiť nejakým polynómom od počtu cifier čísla n (veru tak: polynomiálny od *počtu cifier* čísla n, nie veľkosti čísla n – napr. číslo jedna miliarda je pomerne veľké, ale má iba 10 cifier.)

Algoritmus 2 Implementácia Millerovho testu, ktorý (za predpokladu rozšírenej Riemannovej hypotézy) rozoznáva zložené čísla a prvočísla v čase polynomiálnom od počtu cifier čísla n.

```
def miller (n):
   s, t = get_st (n)
   for i in xrange(2,2*log(n)**2):
     if !strong (n, s, t, a): return False   # zlozene
   return True                               # prvocislo (ERH)
```

☺ *Dobrá správa #2.* Majme nepárne zložené číslo $n \geq 9$. Michael Rabin dokázal, že ak Millerov test spustíme s náhodne vybratým a, pravdepodobnosť, že sa pomýlime a číslo n vyhlásime za prvočíslo je najviac 1/4. (Navyše 1/4 je iba horný odhad – pre väčšinu zložených čísiel je táto pravdepodobnosť ešte menšia.)

To je ale úžasná správa – pre Millerov test neexistuje niečo ako „silné Carmichaelove“ čísla, ktoré by oklamali väčšinu a-čok. Pre každé n ak si zvolíme a náhodne, pravdepodobnosť, že sa pomýlime je najviac $1/4$. Samozrejme, test môžeme niekoľkokrát zopakovať; ak aspoň raz dostaneme odpoveď „zložené číslo“, číslo n je naozaj zložené. Ak vždy dostaneme odpoveď „prvočíslo“, číslo n je asi prvočíslo. Ak test spustíme T-krát, pomýlime sa s pravdepodobnosťou najviac $1/4^T$. Napríklad šanca, že sa pomýlime (t.j. že n je zložené, ale program 50-krát povie „prvočíslo“), ak test spustíme 50-krát s náhodným a, je najviac

$$1/1\,267\,650\,600\,228\,229\,401\,496\,703\,205\,376;$$

slovom: mizivá.

Takéto znáhodnené alebo pravdepodobnostné použitie testu voláme Rabin-Millerov algoritmus; má nasledujúce vlastnosti:

- jeho časová zložitosť je polynomiálna od počtu cifier čísla n;
- algoritmus nie je „deterministický“, ale využíva náhodné čísla;
- algoritmus sa môže mýliť, ale vždy iba „jedným smerom“: môže zložené číslo prehlásiť za prvočíslo (ale nie naopak);
- opakovaním testu môžeme pravdepodobnosť chyby zmenšiť pod ľubovoľne malú hodnotu; presnejšie, ak test zopakujeme T-krát, pravdepodobnosť chyby je najviac $1/4^T$; pre $T = 5$ je to menej ako tisícina, pre $T = 10$ menej ako miliónтina.

Algoritmus 3 Implementácia Miller-Rabinovho testu; program iba 50-krát spustí Millerov test s náhodnými a-čkami; ak povie, že n je zložené, je na 100% zložené; ak povie, že je prvočíslo, môže sa mýliť, ale iba s pravdepodobnosťou menšou ako $1/1\,267\,650\,600\,228\,229\,401\,496\,703\,205\,376$.

```
def miller_rabin (n):
   s, t = get_st (n)
   for i in xrange(0,50):
      a = randint(2, n-1)
      if !strong (n, s, t, a): return False   # zlozene
   return True                                # asi prvocislo
```

☺ *Dobrá správa #3.* Povedali sme si, že žiadna konečná sada a-čiek, ktorými by sme testovali nám na rozhodovanie prvočíselnosti nestačí. Avšak najmenšie číslo, ktoré oklame 2 a 3 zároveň je 1 373 653; najmenšie číslo, ktoré oklame 2 aj 7 aj 61 je 4 759 123 141. V skutočnosti teda, ako vidíme, sú protipríklady dosť veľké. Preto ak chceme testovať iba „malé“ čísla, stačí otestovať niekoľko „vhodných“ a-čiek. Napríklad, ak počítame iba s 32-bitovými číslami[2] (t.j. s

[2] tradičný `unsigned long int` v C či C++ na 32-bitových platformách

číslami do $2^{32} = 4\,294\,967\,296$), stačí urobiť Millerov test so spomínanými $a = 2, 7, 61$. Ďalšie „vhodné“ a-čka sú uvedené v nasledujúcej tabuľke:

pre n menšie ako	stačí otestovať nasledujúce a-čka
1 373 653	$a = 2$ a 3
9 080 191	$a = 31$ a 73
4 759 123 141	$a = 2$, 7 a 61
2 152 302 898 747	$a = 2$, 3, 5, 7 a 11
3 474 749 660 383	$a = 2$, 3, 5, 7, 11 a 13
341 550 071 728 321	$a = 2$, 3, 5, 7, 11, 13 a 17

To, že pre veľa čísiel existuje malá sada a-čiek, ktoré zafungujú, nie je náhoda. Ukážme si, že pre každé n existuje sada $n/2+1$ a-čiek, ktorá funguje pre všetky n-bitové čísla (z tabuľky vidíme, že často stačí ešte menej; napr. pre 32-bitové čísla táto veta hovorí, že existuje sada 17-tich a-čiek, pričom my už vieme, že stačia 3).

Dôkaz. Označme $k = n/2 + 1$. Vieme, že

- všetkých n-bitových čísiel je 2^n a
- ak zvolíme k a-čiek náhodne a vykonáme Millerov test, pomýlime sa s pravdepodobnosťou najviac $1/4^k$ (čo je ostro menej ako $1/2^n$).

Zvoľme teraz k a-čiek náhodne a otestujme úplne všetky n-bitové čísla. Pravdepodobnosť, že sa pomýlime na *aspoň jednom* n-bitovom čísle je $2^n \cdot 1/4^k$, čiže ostro menšia ako 1. To znamená, že sa s nenulovou pravdepodobnosťou nepomýlime na žiadnom n-bitovom čísle. Ale keď tá pravdepodobnosť nie je nulová, tak musí nejaká k-tica a-čiek, pri ktorej sa nepomýlime, existovať. □

Ostáva drobný „problém“: uvedený dôkaz je existenčný – nehovorí nič o tom, ako vhodnú sadu a-čok nájsť, iba že existuje. (Nájsť vhodné a-čka nie je ľahké.)

Algoritmus 4 Implementácia Millerovho testu pre 32-bitové čísla (menšie ako 4 759 123 141).

```
def is_prime32 (n):
   s, t = get_st (n)
   return strong (n, s, t, 2) &&
          strong (n, s, t, 7) &&
          strong (n, s, t, 61)
```