Testovanie prvočíselnosti

```
IS-PRIME(n):
  prime:=true;
  for i:=2 to sqrt(n) do
    if n is divisible by i then
      prime:=false;
  return prime;
```

Časová zložitosť: $O(\sqrt{n})$

Veľkosť vstupu: $x = \log_2 n$ $(\sqrt{n} = 2^{x/2})$

Matematické okienko: Malá Fermatova veta

Ak p je prvočíslo, tak $\forall 0 < a < p : a^{p-1} \equiv 1 (\text{mod } p)$

Príklad:

$2^{12} \equiv 1 (\text{mod } 13) \quad 3^{12} \equiv 1 (\text{mod } 13) \quad 11^{12} \equiv 1 (\text{mod } 13)$

$2^{14} \equiv 4 (\text{mod } 15)$

Fermatov svedok

```
FERMAT-IS-PRIME(n):
  x := 2^(n-1) mod n;
  if (x<>1) return false;
  else return true;
```

... napr. $2^{560} \equiv 1 (\text{mod } 561)$

Matematické okienko: Malá Fermatova veta

Ak p je prvočíslo, tak $\forall 0 < a < p : a^{p-1} \equiv 1 (\text{mod } p)$

Príklad:

$2^{12} \equiv 1 (\text{mod } 13) \quad 3^{12} \equiv 1 (\text{mod } 13) \quad 11^{12} \equiv 1 (\text{mod } 13)$

$2^{14} \equiv 4 (\text{mod } 15)$

```
RANDOM-FERMAT-IS-PRIME(n):
  a := random number between 2 and n-1;
  x := a^(n-1) mod n;
  if (x<>1) return false;
  else return true;
```

. . . Carmichaelove čísla: 561, 1105, 1729, . . .

Matematické okienko: Miller-Rabinova veta

Def (silný svedok): Nech $n - 1 = 2^t u$, kde u je nepárne.

Potom $0 < a < n$ je **silný svedok** akk pre niektoré $0 \leq i < t$:

$a^{2^i . u} \not\equiv \pm 1 (\text{mod } n)$

$a^{2^{i+1} . u} \equiv 1 (\text{mod } n)$

Veta: n je prvočíslo $\Rightarrow$ neexistuje silný svedok

n je zložené $\Rightarrow$ existuje veľa ($\geq \frac{n-1}{2}$) silných svedkov

```
STRONG-WITNESS(a,n):
  decompose n-1 into 2^t.u (u is odd)
  x[0] := a^u mod n;
  for i:=1 to t
    x[i] := x[i-1]^2 mod n;
    if x[i] = 1 and x[i-1]<>1 and x[i-1]<>n-1
    then return true;
  if x[t]<>1 return true;
  return false;

MILLER-RABIN-IS-PRIME(n,s):
  repeat s times
    a := random number between 1 and n-1;
    if STRONG-WITNESS(a,n) return false;
  return true;
```

Časová zložitosť: polynomiálna vzhľadom na $\log n$

```
MILLER-RABIN-IS-PRIME(n,s):
  repeat s times
    a := random number between 2 and n-1;
    if STRONG-WITNESS(a,n) return false;
  return true;
```

- Pravdepodobnostný algoritmus, ktorého **čas behu nezávisí od voľby náhodných čísel**
- Môže sa stať, že nám dá **nesprávnu odpoveď**
  vstup prvočíslo $\Rightarrow$ true
  vstup zložené číslo $\Rightarrow$ môže povedať true
- Nedáva zlú odpoveď systematicky
- Pravdepodobnosť zlej odpovede:
  $\Pr(\text{true}|\text{zlož.číslo}) \leq \Pr(\text{a nie je silný svedok})^s \leq \left(\frac{1}{2}\right)^s$

Monte Carlo algoritmus s jednostrannou chybou

Zložitosť testovania prvočíselnosti

- Miller-Rabinov algoritmus (1980): pravdepobnostný algoritmus v polynomiálnom čase
- Predtým: viaceré rýchle (ale exponenciálne) algoritmy
- Agrawal-Kayal-Saxena (2002): deterministický polynomiálny algoritmus (ale nepraktický / vysoký exponent)

Matematické okienko: Prvočísel je veľa

Našim cieľom je nielen **overovať** ale aj **generovať** náhodné prvočísla.

Veta: Nech $\Pi(n)$ je počet prvočísel $\leq n$. Potom $\lim_{n\to\infty} \frac{\Pi(n)}{n/\ln n} = 1$.

napr. pre $n = 10^9$, $\Pi(n) = 50,847,534$
$n/\ln n = 48,254,942$
(rozdiel menej ako 6%)

```
RANDOM-PRIME(b,s):
  repeat:
    x:=generate random b-bit number
    if MILLER-RABIN-IS-PRIME(x,s) return x
```

Ako dlho bude trvať, kým nájdeme prvočíslo?

```
RANDOM-PRIME(b,s):
  repeat:
    x:=generate random b-bit number
    if MILLER-RABIN-IS-PRIME(x,s) return x
```

$n = 2^b \quad \ln n = b \ln 2$

Šanca, že "trafíme" prvočíslo:

$$\underbrace{\Pr(x \text{ je prvočíslo})}_{p} \approx \frac{n/\ln n}{n} = \frac{1}{\ln n} = \frac{1}{b \ln 2}$$

Cyklíme sa, až kým netrafíme prvočíslo. Aký je počet cyklov?

$n = 2^b \quad \ln n = b \ln 2$

Šanca, že "trafíme" prvočíslo:

$$\underbrace{\Pr(x \text{ je prvočíslo})}_{p} \approx \frac{n/\ln n}{n} = \frac{1}{\ln n} = \frac{1}{b \ln 2}$$

Cyklíme sa, až kým netrafíme prvočíslo. Aký je počet cyklov?

$$T = E[\#\text{ cyklov}] = 1.p + 2.(1-p)p + 3.(1-p)^2 p + \ldots$$

$$\begin{aligned} T - T(1-p) &= p(1 + 2(1-p) + 3(1-p)^2 + 4(1-p)^3 + \ldots) + \\ &\quad p(\quad - (1-p) - 2(1-p)^2 - 3(1-p)^3 - \ldots) \\ &= p(1+(1-p)+(1-p)^2+(1-p)^3+\ldots) = p \cdot \frac{1}{1-(1-p)} = 1 \end{aligned}$$

$$T - T(1-p) = T(1-1+p) = Tp = 1$$

$$T = \frac{1}{p} = \frac{1}{1/b \ln 2} = b \ln 2$$

Očakávaný počet cyklov algoritmu je $b \ln 2$.

(polynomiálny Las Vegas algoritmus kombinovaný s Monte Carlo)

Zhrnutie

- Rabin-Millerov algoritmus na testovanie prvočísel
- tzv. **Monte Carlo** algoritmus:
  - obvykle dáva správnu odpoveď
  - s malou pravdepodobnosťou môže urobiť chybu (o zloženom čísle môže povedať, že je to prvočíslo)
- Prvočísel je veľa
- Vďaka tomu opakovane generovať náhodné číslo a otestovať, či je prvočíslo $\Rightarrow$ **Las Vegas** algorimus.