Identificarea numerelor prime
Etapa 1
Solovay-Strassen și Frobenius

Nonea Victor 325 CD

18.11.2018

De ce sunt relevante numerele prime?

Numerele prime joacă un rol foarte important în sistemele de securitate din moment ce algoritmii de encriptare (gen RSA) își bazează cheile private și publice pe acestea.

Cu cât identifici mai repede numere prime cu atât poți:

1) Să spargi mai repede un strat de securitate bazat pe RSA
2) Să construiești un set de chei de un ordin de mărime cât mai mare, astfel încât cheile private să fie cât mai grele de găsit

Soluțiile alese:

Soluțiile discutate în continuare sunt teste cu caracter **orientativ**. Ele nu determină cu exactitate că un număr este prim, ci doar implică că acel număr este **probabil** prim.

Solovay-Strassen[1]:

Se numește simbol Legendre funcția: $L : N\ X\ P \rightarrow \{0, \pm 1\}$ unde P este mulțimea numerelor prime. *L(a, p)* se notează *(a/p).*

Simbolul Jacobi este o extensie a simbolului Legendre în cadrul căreia *p* nu trebuie să fie neapărat prim.

Algoritmul constă în calcularea simbolului Jacobi pentru un număr oarecare *a*, și *n*, numărul pe care îl testăm, prin diverse proprietăți recursive, și după verificarea dacă valoarea este egală cu cât ar fi fost simbolul Legendre *(a/n)* daca n era prim. (acest rezultat se calculează direct cu formula lui Euler[1])

Frobenius[2]:

Pentru un *n* număr prim, $\forall f : Z_n \to Z_n$, *f(x)* va satisface mai multe proprietăți ce țin de decompunerea sa în factori de grade distincte[2] 2..

Algoritmul constă în verificarea acestor proprietăți pentru un polinom oarecare definit pe Z_n. În implementarea noastră vom genera un polinom cuadratic sau cubic.

Criterii de evaluare:

Pentru testele non-deterministe de numere prime, consider că cel mai important este cât de des este un număr oarecare corect evaluat ca fiind prim; și secundar este cât de repede rulează algorimul.

Voi construi un sample cu toate numerele de la 1 la 10 milioane (cu ciurul lui Eratostene). Voi testa toate valorile cu fiecare din cei doi algoritmi și voi trasa pentru fiecare funcția *nr. evaluate corect / total de nr. evaluate*. Interpolând această funcție aș putea estima acuratețea algoritmilor și pe numere mai mari.

Referințe:

[1]: https://www.geeksforgeeks.org/primality-test-set-4-solovay-strassen/

[2]: http://nntdm.net/papers/nntdm-20/NNTDM-20-4-11-20.pdf