Identificarea numerelor prime

Iordache Tiberiu-Mihai, Grupa 322CD

Universitatea Politehnica din Bucureşti
Facultatea de Automatică şi Calculatoare

Abstract. Acest document este o analiză a algoritmilor de identificare a numerelor prime(Solovay-Strassen şi Frobenius) care prezintă performanţa acestora pe diferite seturi de date.

Keywords: Numere prime · Modulo

1 Descrierea problemei rezolvate

Principala problemă pe care dorim sa o rezolvăm prin intermediul acestui document este identificarea numerlor prime dintr-un set de date de intrare dat.

Această problemă are foarte multe aplicaţii în viaţa de zi cu zi, cea mai întâlnită fiind criptarea datelor. Ştim foarte bine că putem înmulţii două numere prime foarte mari cu uşurinţă, însă procesul invers durează mult mai mult timp, ajutând astfel la o mai bună securizare a datelor noastre.

2 Specificarea soluţiilor alese

Algoritmii aleşi sunt:

2.1 Solovay-Strassen

Acest algoritm determină rapid dacă un număr este compus sau posibil prim. Se folosesc două simboluri:

Simbolul Legendre Fie p un număr impar prim şi a un număr întreg pozitiv, definim simbolul Legendre ca fiind:

$$(\frac{a}{p}) = \begin{cases} 0, & \text{dacă p este un divizor al lui a} \\ +1, & \text{dacă există un } k \text{ întreg astfel încât } k^2 = \text{a(mod p)} \\ -1, & \text{în rest} \end{cases}$$

Datorită unui rezultat al lui Euler, avem relaţia:

$$(\frac{a}{p}) = a^{\frac{p-1}{2}} \text{mod p}$$

Simbolul Jacobian Este o generalizare a simbolului Legendre, unde n este un număr întreg pozitiv:

$$(\frac{a}{n}) = (\frac{a}{p_1})^{k_1} \cdot (\frac{a}{p_2})^{k_2} \cdot ..... \cdot (\frac{a}{p_n})^{k_n}$$

Pentru ca un număr să fie posibil prim, acesta trebuie să aibă simbolul Jacobian egal cu simbolul Legendre.

$$(\frac{a}{n}) = a^{\frac{n-1}{2}} \bmod \text{n}$$

Dacă această condiţie nu este satisfăcută, numărul este compus, iar programul se va oprii. Aşadar, va fi nevoie de mai multe iteraţii pentru a stabili cu un grad mai ridicat de certitudine dacă un număr este prim sau compus.

2.2 Frobenius

Acest algoritm se bazează pe descompunerea unui polinom f(x), definit in Z_n (unde n este numărul căruia dorim să îi testăm primalitatea), în factori de grade distincte. Implementarea se va face prin trei paşi: pasul de factorizare, pasul Frobenius şi pasul Jacobi.

3 Criteriile de evaluare pentru soluţia propusa

Testele vor fi concepute în aşa fel încât să prezinte într-o manieră cât mai obiectivă accurateţea agloritmilor, prin testarea numerelor compuse de diverse valori, şi mari şi mici, dar şi durata de execuţie a acestora, prin testarea unor numere din ce în ce mai mari.

4 Referinţe

1. Solvay-Strassen's Primality Test,
`https://www.geeksforgeeks.org/primality-test-set-4-solovay-strassen/`.
Ultima accesare: 11 Nov 2020.

2. Testing for Prime Numbers: The Solovay-Strassen Algorithm,
`http://www-math.ucdenver.edu/~wcherowi/courses/m5410/ctcprime.html`.
Ultima accesare: 11 Nov 2020.

3. Frobenius pseudoprimes and a cubic primality test,
`http://nntdm.net/papers/nntdm-20/NNTDM-20-4-11-20.pdf`.
Ultima accesare: 11 Nov 2020.