Proiect Analiza Algoritmilor

Baiatu Bianca Daniela
323CA

Universitatea POLITEHNICA Bucuresti
Facultatea de Automatica si Calculatoare
baiatu.daniela2001@gmail.com

Abstract. In cadrul acestei teme am ales sa analizez comparativ 2 algoritmi probabilisti pentru identificarea numerelor prime: Solovay–Strassen si Lucas.

Keywords: Numere prime · Solovay-Strassen · Lucas

1 Introducere

1.1 Descrierea problemei rezolvate

In ciuda faptului ca numerele prime sunt considerate in cele mai multe cazuri derizorii, acestea constituie fundamentul blocurilor tuturor numerelor compuse. Se pune deseori problema descompunerii numerelor in factori primi, idee usor de realizat sau implementat in cazul numerelor mici (10-20 cifre), dar care devine imposibila, spre exemplu, la inmultirea a doua numere prime mari avand fiecare peste 15 cifre. Scopul acestei lucrari este de a compara si exemplifica doi algoritmi non-standard de determinare a numerelor prime avand urmatoarea cerinta: Fiind dat un set de date de intrare se doreste identificarea numerelor prime din acel set.

1.2 Aplicatii practice pentru problema aleasa

O prima utilizare a algoritmilor de identificare a numerelor prime mari, dar si cea mai importanta este in domeniul criptografiei. Astfel, un criptosistem cu cheie publica poate fi folosit pentru criptarea mesajelor astfel incat, in cazul interceptarii, acestea sa nu poata fi decodate. Un algoritm important care se foloseste de numere prime mari este RSA. Alte aplicatii practice ale numerelor prime sunt: generatoarele de numere pseudo-random sau calculul sumelor de control.

1.3 Specificarea solutiilor alese

Pentru rezolvarea problemei propuse am ales compararea testelor de primalitate Solovay-Strassen si Lucas. Testele de primalitate sunt impartite in mai multe categorii: teste deterministice, teste probabilistice, conditionate, neconditionate, teste care functioneaza in timp exponential, subexponential, polinomial. Pentru a diversifica analiza din cadrul proiectului am ales sa examinez doi algoritmi care apartin unor categorii diferite. Solovay-Strassen este un algoritm probabilist, de tip Monte Carlo pentru problema de descompunere in produs de doua numere supraunitare si are o probabilitate de eroare 1/2 [2]. Pe de alta parte, testul Lucas apartine categoriei "Number-theoretic methods", fiind dezvoltat in anul 1891 si sta la baza formularii algoritmilor Agrawal-Kayal-Saxena sau Frobenius [1].

1.4 Criterii de evaluare pentru solutia propusa

Voi compara cei doi algoritmi avand in vedere eficienta fiecaruia la timpul de rulare, complexitatea implementarii acestora, dar si acuratetea rezultatelor obtinute. Setul de teste folosit pentru validarea solutiilor a fost conceput astfel:

- 1. Testele 1 - 5: teste fixe cu un numar redus de elemente (15 - 20) in care sunt testate si cazurile de baza (mai exact, n = 1, n = 2, n = 3). Exista teste care contin in totalitate numere compuse, doar numere prime sau seturi de date combinate.

  - testele 1 - 5: numerele regasite in teste apartin intervalului [0, 100]
- 2. Testele 11 - 20: teste complexe care au scopul de a studia comportamentul celor doi algoritmi alesi in diferite situatii. Astfel, fisierele de input sunt suprascrise la fiecare rulare si contin numere generate random. Toate testele contin proportii egale de numere prime (50%) si numere compuse (50%). Testele sunt construite astfel:
  - testele 6 - 10: contin fiecare cate 2000 de numere. Numerele prime au valoarea mai mare decat PRIME_MAX1, unde PRIME_MAX1 este un macro cu valoarea 10.000. Numerele compuse apartin intervalului [0, NOT_PRIME_MAX1], unde macro-ul NOT_PRIME_MAX1 are valoarea 7.000
  - testele 11 - 15: contin fiecare cate 3000 de numere. Numerele prime au valoarea mai mare decat PRIME_MAX2, unde, unde PRIME_MAX2 este un macro cu valoarea 20.000. Numerele compuse apartin intervalului [0, NOT_PRIME_MAX2], unde macro-ul NOT_PRIME_MAX2 are valoarea 12.000
  - testele 16 - 20: contin fiecare cate 5000 de numere. Numerele prime au valoarea mai mare decat PRIME_MAX3, unde, unde PRIME_MAX3 este un macro cu valoarea 30.000. Numerele compuse apartin intervalului [0, NOT_PRIME_MAX3], unde macro-ul NOT_PRIME_MAX3 are valoarea 20.000

  Observatie: Valorile macro-urilor pot fi schimbate cu usurinta pentru a modifica limitarile intervalelor de numere testate.

Atat fisierele de input, cat si cele de output sunt constuite avand acelasi format:

- pe prima linie se gaseste numarul de numere din fisier, **nr**
- pe urmatoarele **nr** linii se regaseste cate un numar. In cazul fisierelor de input, acestea sunt numere a caror primalitate urmeaza a fi testata, iar in cazul fisierelor de output, acestea au fost identificate ca fiind prime in urma testelor de primalitate.

Generatorul de numere aleatoare

-Se porneste de la o valoare de start verificandu-se primalitatea tuturor numerelor prin metoda trial division. La identificarea unui numar fara divizori proprii, acesta se adauga in vectorul de numere prime.
-Numerele neprime se genereaza random, se verifica primalitatea numarului si daca este compus se adauga in vectorul de numere neprime.
-La final se unesc cei doi vectori, si se apeleaza functia shuffle pentru a amesteca numerele din vectorul final.

References

1. https://www.geeksforgeeks.org/lucas-primality-test/
2. https://www.mta.ro/masterinfosec/files/resources/math.pdf

3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Introduction to Algorithms
4. https://www.geeksforgeeks.org/primality-test-set-4-solovay-strassen/?ref=lbp

Ultima accesare a referintelor: 3 noiembrie 2021