Pirminiai skaičiai

Daumantas Bekintis

Vilniaus universitetas

Matematikos ir informatikos fakultetas

2020-06-05

1. **Uždavinio formuluotė:**

Perinkus natūraliuosius skaičius nuo 1 iki n (kur n – natūralusis skaičius ≥ 2) rasti, kiek tarp jų buvo pirminių skaičių.

Pirminių skaičių radimas yra labai svarbi problema. Jais remiasi šiuolaikinė kriptografija, pvz. RSA kriptosistema, gali būti panaudoti pseudoatsitiktinių skaičių generavimui ir kitoms problemoms.

1. **Algoritmai**:

Pirmasis algoritmas yra **Eratosteno rėtis**. Šis algoritmas buvo žinomas dar antikos laikais. Jo laiko sudėtingumas yra O(n\*log(log(n))), taip pat jis yra tikslus t.y. baigus darbą, jis grąžins visus sekoje esančius pirminius skaičius.

Jo veikimo principas yra labai paprastas: algoritmas prasideda nuo pačio pirmo pirminio skaičiaus – 2 (arba kito, pačio mažiausio pirminio, jei ieškome intervale, kuris neprasideda nuo 1). Algoritmas pažymi visus skaičius, kurie yra dalūs iš dvejeto kaip nepirminius. Tai yra padaroma perėjus per visą duotąją skaičių aibę, kurio žingniu yra lygus einamojo pirminio skaičio dydžiui. Kitaip tariant, jei einamasis pirminis skaičius yra 2, tai jis pažymės antrą, ketvirtą, šeštą ir t.t. kaip ne pirminius. Iteracijai priėjus galą, kitas pirminis skaičius bus likęs mažiausias nepažymėtas skaičius ir algoritmas bus kartojamas analogiškai, kol nebus praeita skaičių

Antrasis algoritmas yra **Rabin – Miller testas pirminiams skaičiams**. Palyginus su Eratosteno rėčiu, jis yra žymiai naujesnis (1976, modifikuotas 1980). Šio algoritmo sudėtingumas O(klog2n), kur k – yra kartojimų skaičius. Taigi, šis algoritmas yra greitesnis nei prieš tai minėtas, bet jis nėra visiškai tikslus – grąžina visus pirminius ir kai kuriuos galbūt pirminius su tikslumu 1-4-k.

Miller – Rabin algoritmas ištiesų neieško, ar skaičius yra pirminis – jis randa ar skaičius yra sudėtinis. Tad skaičiai, kurie lieka po veikimo yra arba pirminiai arba labai stiprūs pseudopirminiai skaičiai. Kaip vyksta šis algoritmas? Štai pavyzdys:

Tarkime, kad ieškome ar skaičius 221 yra pirminis.

1. Randame n-1 = 2km

Užsirašome šį skaičių kaip n – 1 = 220

Daliname iš dviejų iki tol, kol nebegalime ir užrašome kaip 221 – 1 = 22\*55

1. Pasirenkame skaičių a tokį, kad 1 < a < n – 1

Tarkime, tas skaičius yra a = 174

1. Skaičiuojame:

A20d mod n = 17455 mod 221 = 47 ≠ 1, n-1

A21d mod n = 174110 mod 221 = 220 = 1, n-1

Kadangi 220 ≡ -1 mod n, tai gali reikšti, kad 221 yra pirminis (Euklido lema)

Pabandome kitą skaičių a = 137:

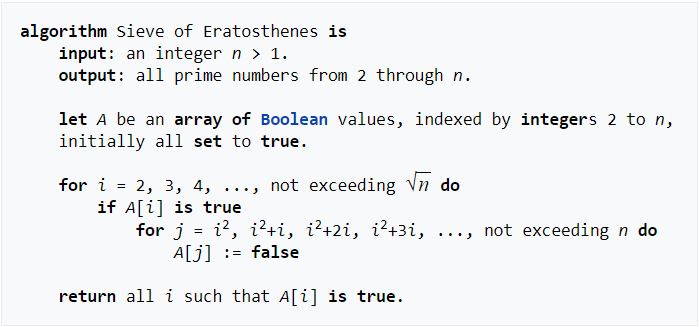
A20d mod n = 13755 mod 221 = 188 ≠ 1, n-1

A21d mod n = 137110 mod 221 = 205 ≠ 1, n-1

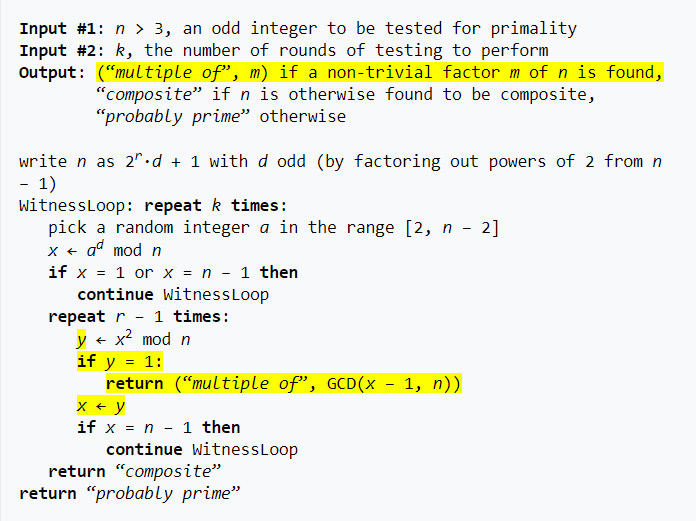
Iš čia randame, kad 221 nėra pirminis, taip pat, kad pakartojimų skaičius padeda sumažinti klaidų skaičių.

**Pseudokodas:**

Eratosteno rėtis



Millerio – Rabino algoritmas



**Eksperimentai**

Visi eksperimentai su paaiškinimais yra įdėti į experiment.txt failą.

Eksperimentų rezultatų vizualizavimas grafikais:

Vykdymo laiko priklausomybė nuo perrenkamų skaičių kiekio:

Tas pats, tik skalė logaritminė:

Iš šių dviejų grafikų matome, kad kartojimų skaičius nedaro labai didelio poveikio algoritmo darbui. Taip pat, laiko sudėtingumas yra aiškiai ne eksponentinis.

Kadangi algoritmas yra netikslus, palyginkime klaidų skaičių su Eratosteno rėčio duomenimis.

Kaip matome, kartojimų skaičius padidina algoritmo tikslumą, tiesa, su mažesniais duomenimis tai nebūtinai būna teisinga.

**Programos naudojimo instrukcija**

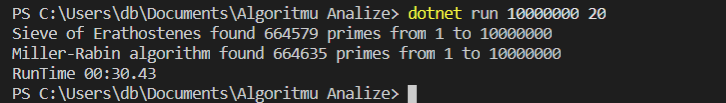
Komandinėje eilutėje nunaviguojame į direktoriją su programos failais. Paleidžiame programą į terminalą įrašę komandą šiuo formatu:

dotnet run l t.

l – natūralusis skaičius, iki kurio programa ieškos pirminių skaičių.

t – natūralusis skaičius, kuris nurodo, kiek kartų Milerio – Rabino algoritmas vykdys pakartojimų

Pavyzdys:



Literatūra:

<https://en.wikipedia.org/wiki/Prime_number>

<https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes>

<https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test>

<https://en.wikipedia.org/wiki/Strong_pseudoprime>

<http://www.allisons.org/ll/AlgDS/Primes/>

<https://en.wikipedia.org/wiki/Fermat%27s_little_theorem>