

Algoritam za pretraživanje prostih brojeva

Znamo li što su prosti brojevi?

Prosti brojevi su prirodni brojevi veći od 1 koji imaju samo dva pozitivna djelitelja: 1 i sami sebe. Drugim riječima, prosti brojevi su brojevi koji su djeljivi samo s 1 i samim sobom, a nemaju druge djelitelje.

Neki od najpoznatijih prostih brojeva su 2, 3, 5, 7, 11, 13, 17, 19, 23, i tako dalje. Primjerice, broj 2 je najmanji prosti broj jer je jedini broj manji od 2 s dva pozitivna djelitelja (1 i 2). Svi ostali prosti brojevi su veći od 2 jer 2 nije djeljiv niti s jednim drugim brojem osim s 1 i samim sobom.

Prosti brojevi su jedan od osnovnih elemenata u teoriji brojeva te imaju važnu ulogu u mnogim matematičkim i računalnim područjima, poput kriptografije, teorije skupova, algoritama i mnogih drugih područja matematike i računarstva.

Prilikom učenja tehnike primjene ponavljanja u algoritmu (petlji) ovo je pronalaženje prostih brojeva u zadanom intervalu je smatram ovo odličnim zadatkom za učenike čijom realizacijom ostvarujemo nekoliko ciljeva nastave matematike/informatike:

- razumijevanje što su prosti brojevi
- usvajanje uvodne matematičke teorije prostih brojeva
- generiranje slučajnih (pseudo) brojeva
- mogući algoritmi provjere prostog broja
- usvajanje algoritamskog rješenja primjenom petlje **for**
- uvođenje kontrolne varijable, ovisno o njejoj vrijednosti ispisujemo rješenje
- primjena naredbe **break** – prekid izvođenja petlje kada više ona nije potrebna
- upoznavanje s funkcijom **perf_counter()** modula **time**
- kritičko promišljanje o efikasnosti pojedinog algoritma

Evo zadatka:

Zadatak 1.

- a) Napisati program u Pythonu koji će generirati nasumičan prirodan broj u segmentu [1, 1000000].
Ispisati poruku je li generirani broj [prost](#) ili nije.
Usporediti brzine izvršavanja algoritama za ispis prostih brojeva u istom programu za generirani broj n .
- b) Program treba ispisati sve proste brojeve do generiranog prirodnog broja n i koliko ima prostih brojeva do generiranog broja n (uključujući njega ukoliko je i sam prost).

Obrazloženje algoritma:

1. Intuitivan način rješenja algoritma za ispitivanje prostog broja je da ispitujemo postoji li djelitelj osim 1 i njega samog. Ukoliko postoji odmah možemo zaključiti da broj nije prost.
2. Teorija matematike dokazuje da najveći djelitelj složenog broja x nikad ne premašuje \sqrt{x} . Kriterij za pronalaženje prostog broja:
Prirodni broj je prost ako i samo ako nije djeljiv ni s jednim prostim brojem koji je manji od ili jednak \sqrt{x} .
3. **Teorem: Svaki se prosti broj, osim brojeva 2 i 3, može zapisati u obliku $6k + 1$ ili $6k - 1$, za $k \in \mathbb{N}$.**

Rješenje:

```
from random import randint
from time import perf_counter
from math import sqrt

br=randint(1,1000000)      #generiramo slučajan broj
print(f'generirani broj: {br}\n')

#Rješenje 1.  ispitujemo postoji li djelitelj do br
t1=perf_counter()
kontrola = True
for i in range(2,br):
    if br%i==0:
        kontrola=False
        break
print('Rješenje 1.:')
if kontrola:
    print('broj',br,'je prost')
else:
    print('broj',br,'nije prost')
t2=perf_counter()
print(f'Vrijeme izvođenja prvog algoritma: {t2-t1}\n')

#Rješenje 2.  ispitujemo postoji djelitelj do sqrt(x)
t1=perf_counter()
kontrola = True
for i in range(2,int(sqrt(br))):
    if br%i==0:
        kontrola=False
        break
print('Rješenje 2.:')
if kontrola:
    print('broj',br,'je prost')
else:
    print('broj',br,'nije prost')
t2=perf_counter()
print(f'Vrijeme izvođenja drugog algoritma: {t2-t1}\n')
```

```
#Rješenje 3. primjenjujemo teorem - rješenje bez ponavljanja u algoritmu
t1=perf_counter()
print('Rješenje 3.:')
if ((br%6==1 or br%6==5) and not(br%5==0)) or br==2 or br==3:
    print('broj',br,'je prost')
else:
    print('broj',br,'nije prost')
t2=perf_counter()
print(f'Vrijeme izvođenja trećeg algoritma: {t2-t1}\n')
```

DZ:

Generirati slučajan broj u intervalu [1, 1000000]. Koristeći jedan od navedenih algoritama ispisati sve proste brojeve do generiranog broja, te koliko ih ima. Ispisivati proste brojeve tablično po 8 u redu. Za svaki broj koristiti 6 mjesta.