PMF Kragujevac
Institut za matematiku i informatiku
Algoritamske strategije, Test
08.11.2013.

Neka su date 2 funkcije, **`prost`** i **`test_prost`**:

```
def prost(broj, delilac):
    if delilac * delilac > broj :
        return True
    else :
        if broj % delilac == 0 :
            return False
        else :
            return prost(broj, delilac + 2)

def test_prost(broj):
    if broj > 2 and broj % 2 == 0 :
        return False
    else :
        return prost(broj, 3)
```

Obe funkcije u sprezi proveravaju da li je broj prost. Funkcija **`test_prost`** prvo proverava da li je broj paran i veći od broja 2. Ako broj nije paran, kontrolu preuzima funkcija **`prost`**, koja proverava da li postoji neparan delilac broja. Granica do koje proverava moguće delioce je prvi neparan broj čiji je kvadrat veći od unetog broja. Prilikom ocene složenosti, za određivanje granice dozvoljeno je koristiti približnu vrednost $\sqrt{broj}$. Za proveru pozivamo npr. **`test_prost(13)`** ili **`test_prost(21)`** i dobijamo respektivno **`True`** tj. **`False`**.

Neka se za jedinicu mere složenosti ovog programa uzme broj izvršavanja komandi u funkcijama. Takođe, neka je funkcija f aritmetička funkcija kojom merimo tu složenost, pri čemu je **`broj`** parametar funkcije. Odrediti za funkciju f(broj):

a) gornju asimptotsku granicu (O) u "najgorem" slučaju i opisati kratko kada se taj slučaj javlja;

b) dati ocenu sa kojom funkcijom ima istu brzinu rasta (θ);

c) (**bonus 2 boda**) u kakvom je odnosu funkcija složenosti f(broj) u odnosu na funkciju složenosti definisanu sa g(broj) = ln(broj) ?

Napomena : izvršavanje komande predstavlja jednu logičku jedinicu posla koji programski kod predstavlja. U prikazanom kodu imamo da funkcija **`test_ prost`** nezavisno od if provere radi 2 koraka (**`if`** proveru i vraćanje vrednosti tj. dalji poziv funkcije **`prost`**). U funkciji **`prost`**, ako je broj deljiv sa nekim deliocem, imamo 3 koraka (2 uzastopne **`if`** provere i vraćanje vrednosti), a ako nije opet imamo 3 koraka (2 provere i rekurzivni poziv).