

## Vježba 5

Cilj tutorijala je upoznavanje sa pojmom rekurzije. Studenti će riješiti nekoliko problema koristeći rekurzivne funkcije, pri čemu će analizirati njihovu vremensku kompleksnost i memorijsko zauzeće.

### Zadatak 1.

U klasu `JednostrukaLista` dodati metodu

```
void ispisUnazad()
```

koja na ekranu ispisuje elemente liste razdvojene jednim znakom razmaka, unazad (od posljednjeg do prvog) koristeći rekurziju. Nije dozvoljeno uništiti listu u procesu.

### Zadatak 2.

Napisati funkciju

```
int binarna_pretraga(Tip* niz, int n, Tip element)
```

koja vraća indeks na kojem se `element` nalazi u `nizu`, odnosno `-1` ako se `element` ne nalazi u `nizu`. Smatrati da je `niz` sortiran u rastućem redoslijedu i da je za `Tip` definiran operator `<`. Očekivana vremenska složenost algoritma je  $O(n)$ .

### Zadatak 3.

Napisati rekurzivnu funkciju

```
double pow(double a, unsigned int n)
```

koja računa  $n$ -ti stepen broja  $a$ . Očekivana vremenska složenost funkcije je  $O(\log n)$ .

### Zadatak 4.

Napisati rekurzivnu funkciju:

```
Stek<Tip> spojiSortirano(Stek<Tip> s1, Stek<Tip> s2)
```

koja prima dva steka čiji su elementi sortirani u rastućem poretku (na vrhu steka je najmanji element), a vraća stek koji sadrži sve elemente iz oba primljena steka (uniju dva steka) čiji su elementi također sortirani u rastućem poretku. Ako se u oba steka nalazi isti element, taj element se u rezultujućem steku treba nalaziti samo jednom.

Funkcija obavezno mora koristiti rekurziju. Očekivana vremenska složenost je  $O(n)$ . Također, očekuje se da rješenje neće koristiti pomoćne stekove osim onog koji se vraća.

### Zadatak 5.

Napisati rekurzivnu funkciju

```
bool daLiJePotencija(int broj, int baza)
```

koja vraća logičku vrijednost `true` (istina) ako je broj **broj** potencija broja **baza**, odnosno ako vrijedi da je  $baza^n = broj$ , gdje je  $n$  neki nenegativan cio broj, a u suprotnom vraća `false`. Npr. broj 8 jeste potencija broja 2 (jer je  $2^3 = 8$ ), ali broj 12 nije.