Bubble sort

Elemente možemo da sortiramo po različitim kriterijuma, u ovoj lekciji ćemo sa $a < b$ proveravati da li element a treba da stoji pre elementa b u sortiranom nizu. Primetimo da znak $<$ ne znači da je a manji broj od b, nego sa time upoređujemo elemente po nekom unapred zadatom kriterijumu. Umesto znaka $<$ može da stoji neka funkcija koja vraća $false$ ili $true$ u zavisnosti da li je element a manji od elementa b po nekom unapred poznatom kriterijumu.

Algoritam bubble sort ili „sortiranje mehurom“ je dobio naziv po tome što elementi poput mehura „isplivaju“ na svoju poziciju. Algoritam se sastoji od n iteracija, gde u svakoj iteraciji zamenimo mesta susednim elementima ukoliko stoje u pogrešnom redosledu. Ukoliko se krećemo kroz niz od prvog do poslednjem elementa posle i-te iteracije će „najvećih“ i elemenata biti sortirani i nalaziće se na pozicijama na kojima treba da se nalaze u sortiranom nizu, a ukoliko se krećemo kroz niz od poslednjeg do prvog elementa posle i-te iteracije će se „najmanjih“ i elemenata nalaziti na prvih i pozicija u nizu i biće dobro raspoređeni.

Pretpostavimo da se krećemo od poslednjeg do prvog elementa u nizu u svakoj iteraciji. U prvoj iteraciji „najmanji“ element će završiti na prvom mestu u nizu, pošto će za njega uvek važiti da je element koji stoji pre njega veći od njega, te će zameniti mesta. Pošto se krećemo nizom od poslednjeg do prvog elementa, „najmanji“ element će menjati mesta sa svim elementima dok ne bude postavljen na prvu poziciju. Posle druge iteracije će drugi „najmanji“ element biti postavljen na drugu poziciju, itd. U i-toj iteraciji će i-ti „najmanji“ element biti postavljen na i-to mesto.

Pseudo kod funkcije možete pogledati u nastavku, gde su parametri niz a koji treba sortirati i broj n koji predstavlja broj elemanata u nizu a.

```
================================================================
01     function BubbleSort( a, n )
02            for iteration = 1 to n do
03                   for i = n-1 to 1 do
04                          if a[i+1] < a[i] then
05                                 swap( a[i+1], a[i] );
06                          end if
07                   end for
08            end for
09     end function
================================================================
```

Funkcija $swap()$ je standardna funkcija koja zamenjuje vrednosti dva elementa.

```
================================================================
01     function Swap( a, b )
02            tmp = a;
03            a = b;
04            b = tmp
05     end function
================================================================
```

Primetimo da nekad nije potrebno da prođemo kroz svih n iteracija. Prvi put kad se desi da u nekoj iteraciji ne zamenimo dva susedna elementa, možemo da zaključimo da je niz sortiran. Spomenuli smo da su posle i-te iteracije prvih i elemenata sortirani, tako da nema potrebe da ponovo prolazimo kroz te elemente. Optimizovan bubble sort možete videti u nastavku.

```
================================================================
01     function BubbleSort( a, n )
02            swapped = true;
03            currentIteration = 1;
04            while ( swapped ) do
05                   swapped = false;
06                   for i = n-1 to currentIteration do
07                          if a[i+1] < a[i] then
08                                 swapped = true;
09                                 swap( a[i+1], a[i] );
10                          end if
11                   end for
12                   currentIteration = currentIteration + 1;
13            end while
14     end function
================================================================
```

Iako je ova verzija bubble sort-a vremenski efikasnija, u najgorem slučaju i dalje mora da odradi n iteracija.