14. Sortiranje niza

Sortiranje niza podrazumeva nalaženje jedne permutacije elemenata niza u kojoj se elementi pojavljuju u neopadajućem tj. nerastućem poretku.

Selection sort

Metoda sortiranja izborom najvećeg elementa odnosi se na sortiranje niza podataka x sa n elemenata u nerastući poredak (slično izbor najmanjeg elementa obezbeđuje sortiranje u neopadajući poredak). Prvo se nalazi najveći element niza i on se "dovodi" na prvo mesto, zatim se nalazi najveći od preostalih n - 1 elemenata i on se "dovodi" na drugo mesto, nalazi najveći od preostalih n-2 elemenata i dovodi na treće mesto, itd, zaključno sa nalaženjem većeg od poslednja dva elementa i njegovim "dovođenjem" na pretposlednje mesto.

Na poslednjem mestu će ostati element koji nije veći ni od jednog u nizu (najmanji element).

Insertion sort

Ako je dat niz (xn) sa elementima nekog, uređenog tipa T, koji treba urediti u neopadajući poredak, ova metoda sortiranja polazi od pretpostavke da imamo uređen početni deo niza, (to svakako važi za i = 2, jer je podniz sa jednim elementom uređen) i u svakom koraku, počevši od i = 2 i povećanjem i, i-ti element se stavlja na pravo mesto u odnosu na prvih (uređenih) i-1.

Bubble sort

Ova metoda je elementarna: ponavlja se prolazak kroz niz elemenata i razmenjuju se susedni elementi, ako je potrebno, sve dok se ne završi prolaz koji ne zahteva nijednu razmenu. Tada je niz sortiran.

Shell sort

Šelsort je jednostavno proširenje sortiranja umetanjem koje dopušta direktnu razmenu udaljenih elemenata. Proširenje se sastoji u tome da se kroz algoritam umetanja prolazi više puta, u prvom prolazu, umesto koraka 1 uzima se neki korak h koji je manji od n (što omogućuje razmenu udaljenih elemenata) i tako se dobija h-sortiran niz, tj. niz u kome su elementi na rastojanju h sortirani, mada susedni elementi to ne moraju biti. U drugom prolazu kroz isti algoritam sprovodi se isti postupak ali za manji korak h. Sa prolazima se nastavlja sve do koraka h = 1, u kome se dobija potpuno sortirani niz.

Quick sort

Ovo je najčešće upotrebljavan algoritam sortiranja. Osnovni oblik algoritma dao je 1960, Hor (Hoare). Nije težak za implementaciju, a koristi manje resursa (vremena i prostora) nego bilo koji drugi algoritam sortiranja, u većini slučajeva. Algoritam ne zahteva dodatnu memoriju, samo n*log(n) operacija u proseku za sortiranje n elemenata, i ima izuzetno kratku unutrašnju petlju. Loše strane algoritma su što je rekurzivan (nerekurzivna varijanta je mnogo složenija), u najgorem slučaju izvršava oko n^2 operacija. Postoje i verzije ovog algoritma koje ga poboljšavaju. Algoritam je vrlo osetljiv na implementaciju (efikasnost se može narušiti lošim izborom u implementaciji). Ako se ne želi analizirati najbolja implementacija, bolje je primeniti šelsort. Ideja algoritma sastoji se u particioniranju niza prema odabranom elementu particioniranja koji se dovodi na pravo mesto, i u primeni algoritma brzog sortiranja na svaku od dve dobijene particije. Rekurzivni poziv se završava kada se primeni na particiju sa manje od dva elementa.

Merge sort

Sortiranje spajanjem ili "merge sort" je algoritam sortiranja zasnovan na poređenju. To je rekurzivni algoritam. Njegova vremenska složenost proporcionalna je sa O(n*log(n)), a u srednjem slučaju je uvek efikasniji od algoritma brzog sortiranja (quick sort). U većini implementacija je stabilan, što znači da zadržava početni redosled jednakih elemenata u sortiranom nizu. Predstavlja primer algoritamske paradigme "podeli pa vladaj". Konstruisao ga je Džon fon Nojman (John von Neumann) 1945. godine. Konceptualno, algoritam sortiranja spajanjem "radi" na sledeći način:

1. Ako niz ima nula ili jedan element, onda je vec soritran. Inače,
2. Podeliti nesortirani niz u dva podniza približno jednake dužine.
3. Sortirati svaki podniz rekurzivno ponovnom primenom algoritma sortiranja spajanjem.

4. Spojiti dva sortirana podniza u jedan sortirani niz.

Algoritam sortiranja spajanjem uključuje dva važna principa kojima poboljšava (smanjuje) vreme izvršavanja:

1. kratki niz je moguće sortirati u manjem broju koraka nego dugački (osnova za deljenje niza na dva podniza)

2. manje koraka je potrebno za konstrukciju sortiranog niza od dva sortirana podniza nego od dva nesortirana podniza (osnova za spajanje).

Sortiranje niza u neopadajućem poretku korišćenjem selection sort-a

```
class SelectionSort
{
public static void main ( String[] args )
        {
        int[] niz = { -20, 19, 1, 5, -1, 27, 19, 5 } ;
        int maksimum, indeks, smena;
        for ( int i=0; i < niz.length; i++ )
                {
                maksimum = niz[i];
                indeks=i;
```

```
            for ( int j=i+1; j< niz.length; j++ )
                    {
                    if ( niz[j] > maksimum )
                            {
                            maksimum = niz[ j ];
                            indeks=j;
                            }
                    }
            smena=niz[i];
            niz[i]=niz[indeks];
            niz[indeks]=smena;
            }
    }
}
```

```
class SelectionSort1
{
public static void main ( String[] args )
        {
        int[] niz = { -20, 19, 1, 5, -1, 27, 19, 5 } ;
        int smena;
        for ( int i=0; i < niz.length; i++ )
                {
                for ( int j=i+1; j< niz.length; j++ )
                        {
                        if (niz[i] < niz[j] )
                                {
                                smena=niz[i];
                                niz[i]=niz[j];
                                niz[j]=smena;
                                }
                        }
                }
        }
```

Kontrolna pitanja

59. Nabrojte algoritme za sortiranje niza.
60. Opišite postupak selection sort.
61. Opišite postupak insertion sort.
62. Opišite postupak bubble sort.
63. Opišite postupak shell sort.
64. Opišite postupak quick sort.
65. Opišite postupak merge sort.

Kontrolna pitanja

66. Napravite program koji sortira neki niz u neopadajućem poretku.