

Programiranje i programski jezici



Algoritmi za sortiranje nizova

Sortiranje niza celih brojeva u neopadajući poredak

```
#include <stdio.h>
ALGORITAM ZA SORTIRANJE
main(){
    int i,n,a[50];
    scanf("%d",&n);
    for(i=0; i<n; i++) scanf("%d",&a[i]);
    POZIV ALGORITMA ZA SORTIRANJE
    for(i=0; i<n; i++) printf("%5d",a[i]);
    printf("\n");
}
```

Insertion-sort

Sortiranje umetanjem elemenata niza na odgovarajuća mesta

- Nakon k iteracija pri sortiranju:
 - Na početku niza se nalazi sortirani deo niza
 - Ostatak niza je deo koji tek treba sortirati
- Novi element se dodaje na kraj sortiranog dela niza i zatim se poredi sa prethodnim elementom (ako postoji)
- Ako je novi element manji, oni zamenjuju mesta
- Postupak se ponavlja sve dok postoji prethodnik koji je veći.

Insertion-sort

```
void isort (int n, int a[]) {  
    int i,j,k;  
    for(i=0; i<n; i++)  
        for(j=i; (j>0) && (a[j]<a[j-1]); j--) {  
            k=a[j]; a[j]=a[j-1]; a[j-1]=k;  
        }  
}
```

```
void isort2(int n, int a[]){  
    int i,j,k;  
    for(i=0;i<n;i++){  
        k=a[i];  
        for(j=i-1;(j>=0)&&(a[j]>k);j--) a[j+1]=a[j];  
        a[j+1]=k;  
    }  
}
```

Insertion-sort

Za $n=8$ i niz 44, 55, 12, 42, 94, 18, 6, 67 dobija se:

44	55	12	42	94	18	6	67
44	55	12	42	94	18	6	67
12	44	55	42	94	18	6	67
12	42	44	55	94	18	6	67
12	42	44	55	94	18	6	67
12	18	42	44	55	94	6	67
6	12	18	42	44	55	94	67
6	12	18	42	44	55	67	94

Selection-sort

Sortiranje izborom uzastopnih minimuma

- Uređuje niz u rastući tako što traži najmanji element i smešta ga na početak niza.
- Zatim se za ostatak niza ponavlja isti postupak, sve dok se ne dođe do podniza koji ima samo jedan element.

Selection-sort

```
void ssort(int a[],int n){  
    int i,j,pom;  
    for (i=0;i<n-1;i++) {  
        for(j=i+1;j<n;j++)  
            if (a[j]<a[i]) {  
                pom=a[i];  
                a[i]=a[j];  
                a[j]=pom;  
            }  
        }  
    }  
}
```

Bubble-sort

Sortiranje poređenjem parova uzastopnih elemenata

- Porede se svaka dva uzastopna elementa niza
- Ako je u nekom paru prethodnik veći od sledbenika, ta dva elementa zamenjuju mesta
- Ovakvim poređenjem će na poslednje mesto, u prvom prolazu biti postavljen najveći element niza
- Postupak se ponavlja dok se ne uporede svi parovi

Bubble-sort

```
void bubble(int n, int a[]){
    int i,j,k;
    for(i=n-1; i>0; i--){
        for(j=0; j<i; j++){
            if (a[j]>a[j+1]){
                k=a[j]; a[j]=a[j+1]; a[j+1]=k;
            }
        }
    }
}
```

```
void bubble2(int n, int a[]){
    int i,j,k,swp=1;
    for(i=n-1; swp && (i>0); i--){
        swp=0;
        for(j=0; j<i; j++){
            if (a[j]>a[j+1]){
                k=a[j]; a[j]=a[j+1]; a[j+1]=k;
                swp=1;
            }
        }
    }
}
```

Bubble-sort

Za $n=8$ i niz 44, 55, 12, 42, 94, 18, 6, 67 dobija se:

44	12	42	55	18	6	67	94
12	42	44	18	6	55	67	94
12	42	18	6	44	55	67	94
12	18	6	42	44	55	67	94
12	6	18	42	44	55	67	94
6	12	18	42	44	55	67	94
6	12	18	42	44	55	67	94

Merge-sort

Sortiranje spajanjem sortiranih nizova

- Metod pogodan za “spoljašnje” sortiranje – sortiranje na spoljašnjem memorijskom medijumu
- Par uzastopnih najdužih neopadajućih podnizova se spaja (uparuje) i dobija se jedan neopadajući podniz
- Svakim prolazom se smanjuje broj neopadajućih podnizova

Merge-sort

```
void merge_arr(int a1,int a2,int b1,int b2,int *c,int *d){
    int k1;
    k1=a1;
    while ((a1<=a2) && (b1<=b2))
        if (c[a1]<c[b1]) d[k1++]=c[a1++];
        else d[k1++]=c[b1++];
    while (a1<=a2) d[k1++]=c[a1++];
    while (b1<=b2) d[k1++]=c[b1++];
}
```

```
void dodeli(int p,int k, int a[], int b[]){
    int i;
    for(i=p;i<=k;i++) a[i]=b[i];
}
```

Merge-sort

```
void merge(int n, int na[]){
    int a1=0,a2,b1,b2=-1,k=1,nb[n],i;
    a1=a2=0;
    while ((na[a2]<na[a2+1])&&(a2<n-1)) a2++;
    while(k){
        if((b1=b2=a2+1)==n) k=0;
        else {
            while ((na[b2]<na[b2+1])&&(b2<n-1)) b2++;
            merge_arr(a1,a2,b1,b2,na,nb);
            dodeli(a1,b2,na,nb);
            a2=b2;
        }
    }
}
```

Merge-sort

Za $n=8$ i niz 44, 55, 12, 42, 94, 18, 6, 67 dobija se:

44	55	12	42	94	18	6	67
12	42	44	55	94	18	6	67
12	18	42	44	55	94	6	67
6	12	18	42	44	55	67	94

Merge-sort

```
void merge2(int begin,int end,int a[]){  
    int mid,temp[50];  
    if (begin<end) {  
        mid=(begin+end)/2;  
        merge2(begin,mid,a);  
        merge2(mid+1,end,a);  
        merge_arr(begin,mid,mid+1,end,a,temp);  
        dodeli(begin,end,a,temp);  
    }  
}
```

Merge-sort

Za $n=8$ i niz 44, 55, 12, 42, 94, 18, 6, 67 dobija se:

44	55	12	42
----	----	----	----

94	18	6	67
----	----	---	----

44	55
----	----

12	42
----	----

94	18
----	----

6	67
---	----

44

55

12

42

94

18

6

67

44	55
----	----

12	42
----	----

18	94
----	----

6	67
---	----

12	42	44	55
----	----	----	----

6	18	67	94
---	----	----	----

6 12 18 42 44 55 67 94

Stack-sort

Sortiranje korišćenjem steka

- Pri sortiranju se koriste dva steka - jednodimenzioni niz
- Elementi se dodaju u jedan od stekova tako da važi
 - u prvom steku elementi su od dna ka vrhu poređani u neopadajućem redosledu
 - u drugom steku je obratno
 - u drugom steku svi elementi su veći od elemenata iz prvog steka
- Dodavanje elemenata prebacivanjem tako da
 - na vrhu prvog steka je element manji ili jednak elementu koji se dodaje
 - na vrhu prvog steka je element veći ili jednak elementu koji se dodaje

Stack-sort

```
void stack(int n, int a[]){
    int stm[MAXEL], stv[MAXEL];
    int vrm, vrj, i, j;
    vrm=vrj=-1;
    for(i=0; i<n; i++){
        while((vrm!=-1) && (stm[vrm]>a[i]))
            stv[++vrj]=stm[vrm--];
        while((vrj!=-1) && (stv[vrj]<a[i]))
            stm[++vrm]=stv[vrj--];
        stm[++vrm]=a[i];
    }
    for(i=vrm; i>=0; i--) a[i]=stm[i];
    for(i=vrj, j=vrm+1; i>=0; i--) a[j++]=stv[i];
}
```

Stack-sort

Za $n=8$ i niz 44, 55, 12, 42, 94, 18, 6, 67 vrednosti na stekovima su:

Manji stek	44							
Veci stek								
Manji stek	44	55						
Veci stek								
Manji stek	12							
Veci stek	55	44						
Manji stek	12	42						
Veci stek	55	44						
Manji stek	12	42	44	55	94			
Veci stek								
Manji stek	12	18						
Veci stek	94	55	44	42				
Manji stek	6							
Veci stek	94	55	44	42	18	12		
Manji stek	6	12	18	42	44	55	67	
Veci stek	94							

Quick-sort - Sortiranje deljenjem niza

Osnovna ideja je da se niz podeli na dve grupe elemenata, grupu elemenata koji su manji ili jednaki izabranom elementu niza, i grupu elemenata koji su veći od izabranog elementa, između ove dve grupe nalazi se izabrani element. Zatim se za svaku od grupa ponavlja postupak. Podela niza obično se vrši u odnosu na prvi, poslednji ili srednji element niza. U rešenju koje sledi deljenje niza se vrši u odnosu na prvi element niza.

Podela niza u grupe vrši se tako što redom analiziramo elemente niza i dodajemo ih odgovarajućoj grupi. Promenljivom k pamtimo granicu između dve grupe. Kako podelu vršimo u odnosu na prvi element k postavimo na indeks prvog elementa niza. U momentu kad analiziramo i -ti član niza

- elementi od početka niza do k -tog elementa (uključujući i k -ti) pripadaju grupi elemenata koji su manji ili jednaki
- elementi od $(k+1)$ -og do $(i-1)$ -og pripadaju grupi elemenata koji su veći
- elementi od i -tog do kraja niza nisu još obrađeni.

Quick-sort - Sortiranje deljenjem niza

Ako element $a[i]$ pripada grupi manjih ili jednakih dodajemo ga u tu grupu, tako što razmenimo mesta elementu $a[i]$ i prvom elementu iz grupe većih $a[k+1]$, i pri tome uvećamo grupu manjih ili jednakih za 1 ($k++$).

Nakon analize svih elemenata niz se sastoji iz samo dve grupe

- elementi od početka niza do k -tog elementa (uključujući i k -ti) pripadaju grupi elemenata koji su manji ili jednaki prvom elementu niza
- elementi od $(k+1)$ -og do poslednjeg elementa niza pripadaju grupi elemenata koji su veći od prvo elementa niza.

Na kraju razmenimo početni element niza i $a[k]$ da bi granični element ($a[k]$) bio između grupe manjih ili jednakih i grupe većih, na taj način element $a[k]$ je postavljen na pravo mesto u nizu (levo od njega su manji ili jednaki, a desno veći). Još je preostalo da se istim postupkom sortira deo niza od početka do $(k-1)$ -og člana, i deo niza od $(k+1)$ -og do kraja niza.

Quick-sort

```
void zameni(int a[],int i,int j){
    int pom;
    pom=a[i];
    a[i]=a[j];
    a[j]=pom;
}
int podeli(int a[],int donji,int gornji) {
    int i,pivot,k;
    pivot=donji;
    for(i=donji+1;i<=gornji;i++)
        if(a[i]<a[donji]){
            pivot++;
            zameni(a,i,pivot);
        }
    zameni(a,donji,pivot);
    for(k=donji; k<=gornji; k++) printf("%5d",a[k]);
    printf("\n");
    return(pivot);
}
```

Quick-sort

```
void quick(int a[],int donji,int gornji){
    int pivot,k;
    if(donji>=gornji) return;
    if(donji+1==gornji){
        if(a[donji]>a[gornji]) zameni(a,donji,gornji);
        return;
    }
    //for(k=donji; k<=gornji; k++) printf("%5d",a[k]);
    //printf("\n");
    pivot =podeli(a,donji,gornji);
    quick(a,donji,pivot-1);
    quick(a,pivot+1,gornji);
}
```

Quick-sort

7	4	9	10	1	2	16	8	3	14
---	---	---	----	---	---	----	---	---	----

7	4	9	10	1	2	16	8	3	14
---	---	---	----	---	---	----	---	---	----

7	4	9	10	1	2	16	8	3	14
---	---	---	----	---	---	----	---	---	----

7	4	9	10	1	2	16	8	3	14
---	---	---	----	---	---	----	---	---	----

7	4	1	10	9	2	16	8	3	14
---	---	---	----	---	---	----	---	---	----

7	4	1	2	9	10	16	8	3	14
---	---	---	---	---	----	----	---	---	----

7	4	1	2	9	10	16	8	3	14
---	---	---	---	---	----	----	---	---	----

7	4	1	2	9	10	16	8	3	14
---	---	---	---	---	----	----	---	---	----

7	4	1	2	3	10	16	8	9	14
---	---	---	---	---	----	----	---	---	----

7	4	1	2	3	10	16	8	9	14
---	---	---	---	---	----	----	---	---	----

3	4	1	2	7	10	16	8	9	14
---	---	---	---	---	----	----	---	---	----