

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;  
        }  
}
```

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

Insertion-sort – ver. 2

```
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;
    }
}
```

Bubble-sort

Sortiranje poredjenjem parova uzastopnih elemenata

- Porede se svaka sva uzastopne elementa niza
- Ako je u nekom paru prethodnik veći od sledbenika, ta dva elementa zamenjuju mesta
- Ovakvim poredjenjem će na posledenje 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;
            }
}
```

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

Bubble-sort – ver. 2

```
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;
            }
        }
    }
}
```

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,a2,b1,b2=-1,k=1,nb[n];
    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 – ver. 2

```
void merge2(int begin,int end,int a[]){
    int mid,temp[50];
    if ((begin!=end)&&(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[n],stv[n];
    int vrm,vrv,i,j;
    vrm=vrv=-1;
    for(i=0;i<n;i++){
        while((vrm!=-1) && (stm[vrm]>a[i]))
            stv[++vrv]=stm[vrm--];
        while((vrv!=-1) && (stv[vrv]<a[i]))
            stm[++vrm]=stv[vrv--];
        stm[++vrm]=a[i];
    }
    for(i=vrm;i>=0;i--) a[i]=stm[i];
    for(i=vrv,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							