

# Сортиране на масиви



# Размяна на местата на два елемента на масив

1. `int swap;`
2. `swap=a[i];`
3. `a[i]=a[j];`
4. `a[j]=swap;`

# Отпечатване на масив

1. `for (int i=0; i<n; i++)`
2. `cout<<a[i]<<' ';`
3. `cout<<endl;`

# Метод на пряката селекция - действие

1. Обхождаме целия масив и намираме най-малкия елемент
2. Записваме го на **първа** позиция, а **първият** елемент – на неговото място
3. Обхождаме целия останал масив и намираме най-малкия елемент
4. Записваме го на **втора** позиция, а **вторият** елемент – на неговото място
5. ...
6. Обхождаме останалият масив и намираме най-малкия елемент
7. Записваме го на позиция  **$n-1$** , а елемент  **$n-1$**  – на неговото място

# Метод на пряката селекция - код

```
1.  int swap, imin;
2.  for (int i=0; i<n-1; i++) {
3.      imin=i;
4.      for (int j=i+1; j<n; j++)
5.          if (a[j]<a[imin]) imin=j;
6.      if (imin != i) {
7.          swap=a[i];
8.          a[i]=a[imin];
9.          a[imin]=swap;
10.     }
11. }
```

# Метод на пряката размяна (метод на мехурчето) - действие

1. Сравняваме първия и втория елемент
2. Ако първия е по-голям от втория, им разменяме местата
3. Повтаряме същото за втория и третия, третия и четвъртия и т.н.
4. Така най-големия елемент става последен
5. Повтаряме същото за всички елементи от 1-вия до  $n-1$ -вия, после до  $n-2$ -рия и т.н.

# Метод на пряката размяна (метод на мехурчето)

```
1.  int swap;  
2.  for (int i=0; i<n-1; i++)  
3.      for (int j=0; j<n-i-1; j++)  
4.          if (a[j]>a[j+1]) {  
5.              swap=a[j];  
6.              a[j]=a[j+1];  
7.              a[j+1]=swap;  
8.          }  
9.  for (int i=0; i<n; i++)  
10.     cout<<a[i]<<' '  
11. cout<<endl;
```

# Край



на масивите