



Методы сортировки массива

Определение



Сортировкой или **упорядочением** массива называется расположение его элементов по возрастанию (или убыванию). Если не все элементы различны, то говорят о *неубывающем* (или *невозрастающем*) порядке.



Цитаты великих людей

"Даже если бы сортировка была почти бесполезна, нашлась бы масса причин заняться ею! Изобретательные методы сортировки говорят о том, что она и сама по себе интересна как объект исследования."

/Д. Кнут/

"Создается впечатление, что можно построить целый курс программирования, выбирая примеры только из задач сортировки."

/Н. Вирт/

Методы сортировки массива

- 1. сортировка вставкой (включением);**
- 2. сортировка выбором (выделением);**
- 3. сортировка обменом ("пузырьковая"
сортировка).**

Сортировка выбором

Принцип метода:

Находим (выбираем) в массиве элемент с минимальным значением на интервале от 1-го элемента до n -го (последнего) элемента и меняем его местами с первым элементом. На втором шаге находим элемент с минимальным значением на интервале от 2-го до n -го элемента и меняем его местами со вторым элементом. И так далее для всех элементов до $(n-1)$ -го.

Отсортировать массив в порядке возрастания (метод выбора)

```
var a:array [1..6] of integer; i,j,Min,MinI:integer;
Begin
  for i:=1 to 6 do
    begin
      write ('a[' ,i,']=');
      readln (a[i]);
    end;
  for i:=1 to 6 do
    begin
      Min:=a[i];
      MinI:=i;
      for j:=i+1 to 6 do
        if a[j]<Min then begin Min:=a[j]; MinI:=j;end;
      a[MinI]:=a[i];
      a[i]:=Min;
    end;
  for i:=1 to 6 do
    write(a[i], ' ');
End.
```

Сортировка методом вставки

Принцип метода:

Массив разделяется на две части: отсортированную и не отсортированную. Элементы из не отсортированной части поочередно выбираются и вставляются в отсортированную часть так, чтобы не нарушить в ней упорядоченность элементов. В начале работы алгоритма в качестве отсортированной части массива принимают только первый элемент, а в качестве не отсортированной - все остальные элементы.

Алгоритм:

Алгоритм будет состоять из $(n-1)$ -го прохода (n - размерность массива), каждый из которых будет включать четыре действия:

- 1) взятие очередного i -го не отсортированного элемента и сохранение его в дополнительной переменной;
- 2) поиск позиции j в отсортированной части массива, в которой присутствие взятого элемента не нарушит упорядоченности элементов;
- 3) сдвиг элементов массива от i -го до $j-1$ -го вправо, чтобы освободить найденную позицию вставки;
- 4) вставка взятого элемента в найденную i -ю позицию.

Отсортировать массив в порядке возрастания (метод вставки).

```
Var i,j,e,g:integer; a:array [1..6] of integer;
```

```
Begin
```

```
  for i:=1 to 6 do
```

```
  begin
```

```
    write ('a[' ,i,']=');
```

```
    readln (a[i]);
```

```
  end;
```

```
  for i:=2 to 6 do
```

```
  begin
```

```
    e:=A[i];
```

```
    j:=1;
```

```
    while (e>a[j]) do
```

```
      Inc(j);
```

```
    for g:=i-1 downto j do
```

```
      a[g+1]:=a[g];
```

```
      a[j]:=e;
```

```
  end;
```

```
  for i:=1 to 6 do
```

```
    write(a[i], ' ');
```

```
End.
```

Сортировка методом «пузырька»

Принцип метода:

В сортировке методом пузырька по возрастанию более легкие (с меньшим значением) элементы постепенно "всплывают" в начало массива, а более тяжелые друг за другом опускаются на дно (в конец массива).

Алгоритм:

Элементы попарно сравниваются между собой: первый со вторым, затем второй с третьим, следом третий с четвертым и т.д. Если предшествующий элемент оказывается больше последующего, то их меняют местами. Постепенно самое большое число оказывается последним.

Отсортировать массив в порядке возрастания (метод «пузырька»)

```
var a: array[1..6] of integer; i, j, k: integer;
begin
  for i:=1 to 6 do
    begin
      write ('a[' , i, ']=');
      readln (a[i]);
    end;
    for i := 1 to 5 do
      for j := 1 to 5 do
        if a[j] > a[j+1] then begin
          k := a[j];
          a[j] := a[j+1];
          a[j+1] := k
        end;
      for i := 1 to 6 do
        write (a[i], ' ');
      end.
```