



МАССИВЫ

1. Методы сортировки.

Сортировка методом

простого включения

Алгоритм: (на примере сортировки по убыванию)

1. На k -ом шаге считаем, что часть массива, содержащая первые $k-1$ элементов уже упорядочена, то есть $a[1] \geq a[2] \geq \dots \geq a[k-1]$
2. Берем k -ый элемент и подбираем для него место в отсортированном массиве такое, чтобы после его вставки упорядоченность не нарушилась. То есть необходимо найти j , которое удовлетворяло бы условиям: $1 \leq j \leq k-1$, $a[j] \geq a[k] \geq a[j+1]$
3. Вставляем элемент $a[k]$ на найденное место.



```
for k := 2 to n do
```

```
begin
```

```
  x := a[k];
```

```
  {вставить x на подходящее место в
```

```
  a[1] >= a[2] >= ... >= a [k-1]}
```

```
end;
```

Как найти подходящее место для X?

Алгоритм:

1. Просматриваем элементы массива (упорядоченного), двигаясь от конца к началу массива (то есть от $k-1$ до 1)
2. Просматриваем пока не будет выполнено одно из условий:
 - a) найдем $a[j] < x$ (будем вставлять между $a[j-1]$ и $a[j]$)
 - b) достигнут левый конец упорядоченной части массива (тогда необходимо x вставить на 1-е место)
3. Пока условия 2 не выполнены будем смещать просматриваемые элементы на 1 позицию вправо, в результате чего в отсортированной части будет освобождено место под X .



for $k := 2$ **to** n **do**

begin

$x := a[k]; j := k-1;$

while $(j > 0)$ **and** $(x \geq a[j])$ **do**

begin


$a[j+1] := a[j]; j := j - 1;$


end;

$a[j+1] := x$

end;

1. Будет ли сортировка выполняться правильно, если в заголовке цикла **while** указать $x > a[j]$?
2. Сколько при данном методе сортировки производится сравнений в лучшем и худшем случаях?
3. В алгоритме сортировки массива необходимо было искать место вставки очередного элемента в отсортированную часть. Использование для этого бинарного поиска позволяет значительно улучшить степень эффективности сортировки. Такой модифицированный алгоритм сортировки называют методом бинарного включения. Напишите программу, реализующую этот метод.

- 
1. Да. Просто равные элементы будут вставляться не до соответствующего равного, а после.
 2. от $n-1$ до $n \cdot (n-1) / 2$



for $i := 2$ to n **do**

if $a[i-1] > a[i]$ **then begin**

$x := a[i]$; $left := 1$; $right := i-1$;

repeat

$m := (left + right) \text{div } 2$;

if $a[m] < x$ **then**

$left := m+1$ **else** $right := m-1$; **until**

$left > right$;

for $j := i-1$ **downto** $left$ **do**

$a[j+1] := a[j]$; $a[left] := x$;

end;

