

## СРСП-10

### ТЕМА: УПОРЯДОЧИВАНИЕ (СОРТИРОВКА) МАССИВОВ

Цель работы: *в данной лабораторной работе необходимо изучить ряд известных алгоритмов сортировки и создать комплекс программ, реализующий*

- линейный поиск элемента в массиве;
- метод двоичного поиска элемента в упорядоченном массиве;

Программа должна обеспечить сортировку массивов размером произвольной длины до 10000 элементов и выводить для контроля:

- при небольшом количестве элементов (например, менее 20) - неупорядоченный массив и массив после сортировки для каждого из предложенных алгоритмов;

- при значительном объеме данных (более 20) выводить время сортировки одного и того же массива для всех четырех предложенных алгоритмов.

Содержимое массива рекомендуется формировать с помощью генератора псевдослучайных чисел, замеры времени производить средствами модуля System (Random, DateTime).

В ходе выполнения лабораторной работы предполагается реализация методов *линейного* и *двоичного поиска* элемента массива.

Результатом работы алгоритма линейного поиска значения Val в массиве A являются индекс Pos и логическая переменная ResultOk, которая принимает значение TRUE, если такой элемент содержится в массиве A, и FALSE - в противном случае. Индекс Pos принимает значение, равное номеру искомого элемента, если такой найден, и значение, равное -1 - в противном случае.

Алгоритм линейного поиска

Шаг 1. Полагается Pos:=-1 и ResultOk:=FALSE, и значение переменной цикла J:=0.

Шаг 2. Если A[J]=Val, то переменным Pos и ResultOk присваиваются соответственно значения Pos:=J, ResultOk:=TRUE и алгоритм завершает работу. В противном случае значение переменной цикла увеличивается на единицу J:=J+1.

Шаг 3. Если J<Last, где Last - число элементов массива A, то выполняется Шаг 2, в противном случае - работа алгоритма завершена.  
Конец алгоритма.

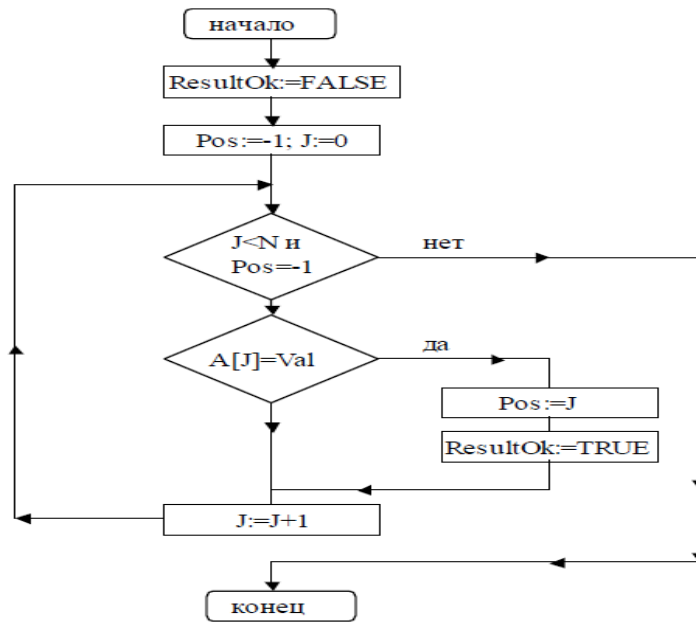


Рис 10.1. Блок-схема алгоритма линейного поиска

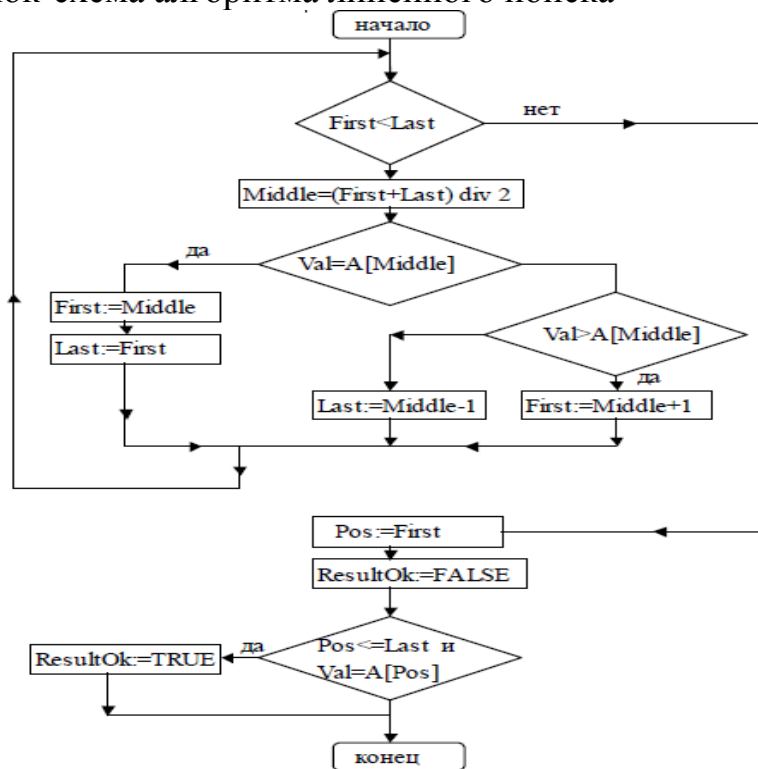


Рис10.2. Блок-схема алгоритма двоичного поиска

### 10.3 Метод двоичного поиска

Результатом работы алгоритма является индекс Pos, показывающий на место в упорядоченном массиве A с номерами элементов от First до Last, на которое необходимо поставить значение Val так, чтобы вновь образованный массив остался упорядоченным. Формируется в качестве результата и логическая переменная ResultOk, которая принимает значение TRUE, если искомый элемент содержится в массиве A, и - FALSE - в противном случае.

### Алгоритм двоичного поиска

Шаг 1. Пока справедливо условие  $First < Last$ , выполняется Шаг 2.

Шаг 2. Вычисляется середина массива  $Middle := (Last + First) \div 2$ . Если  $Val$  равно  $A[Middle]$ , то полагается  $First := Middle$  и  $Last := First$ , в противном случае проверяется условие -  $Val$  больше  $A[Middle]$ ? Если это условие справедливо, то полагается  $First := Middle + 1$ , в противном случае полагается  $Last := Middle - 1$ . После чего управление передается на Шаг 1.

Шаг 3. Полагается  $Pos := First$ .

Шаг 4. Проверка успеха поиска элемента  $Val$  в массиве. Полагается  $ResultOk := FALSE$ . После чего проверяется, содержится ли элемент со значением  $Val$  в массиве, и при положительном ответе на этот вопрос переменной  $ResultOk$  присваивается значение  $TRUE$ .

Конец алгоритма.