

**УВАЖАЕМЫЕ СТУДЕНТЫ!
ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:**

1. Ознакомиться с теорией и законспектировать лекцию не меньше трех листов, составить и ответить на вопросы.
2. Предоставит отчет конспекта лекции прислать в виде скриншото в течении трех дней.
3. Отправить преподавателю на почту v.vika2014@mail.ru и указать свою Ф.И.О, группу, и название дисциплины тел 072-17-44-9-22

Тема: Методы перебора в задачах поиска.

Задачи поиска предназначены для определения нахождения элемента, обладающего заданным свойством, в определенной совокупности данных, в частности, в массиве.

Линейный поиск.

Поиск наибольшего и наименьшего элемента в массиве.

Дан ряд чисел $a_1, a_2, \dots, a_i, \dots, a_n$. Разработать алгоритм поиска наибольшего и наименьшего числа в этом ряду с указанием номера (индекса) его расположения.

Очевидный способ поиска наибольшего (наименьшего) числа в заданном ряду n чисел включает следующие действия. Взять первое число ряда и сказать, что оно наибольшее (наименьшее), а индекс его равен 1. Затем взять второе число ряда и сравнить с предполагаемым максимальным (минимальным) первым числом. Если второе число больше предполагаемого (максимального) первого числа, взять третье число ряда и сравнить с первым. Так следует действовать до тех пор, пока не будет выбрано последнее a_n число. В результате на месте первого числа окажется наибольшее (наименьшее) число с указанным его номером в ряду чисел.

Блок – схема алгоритма поиска наибольшего и наименьшего элемента

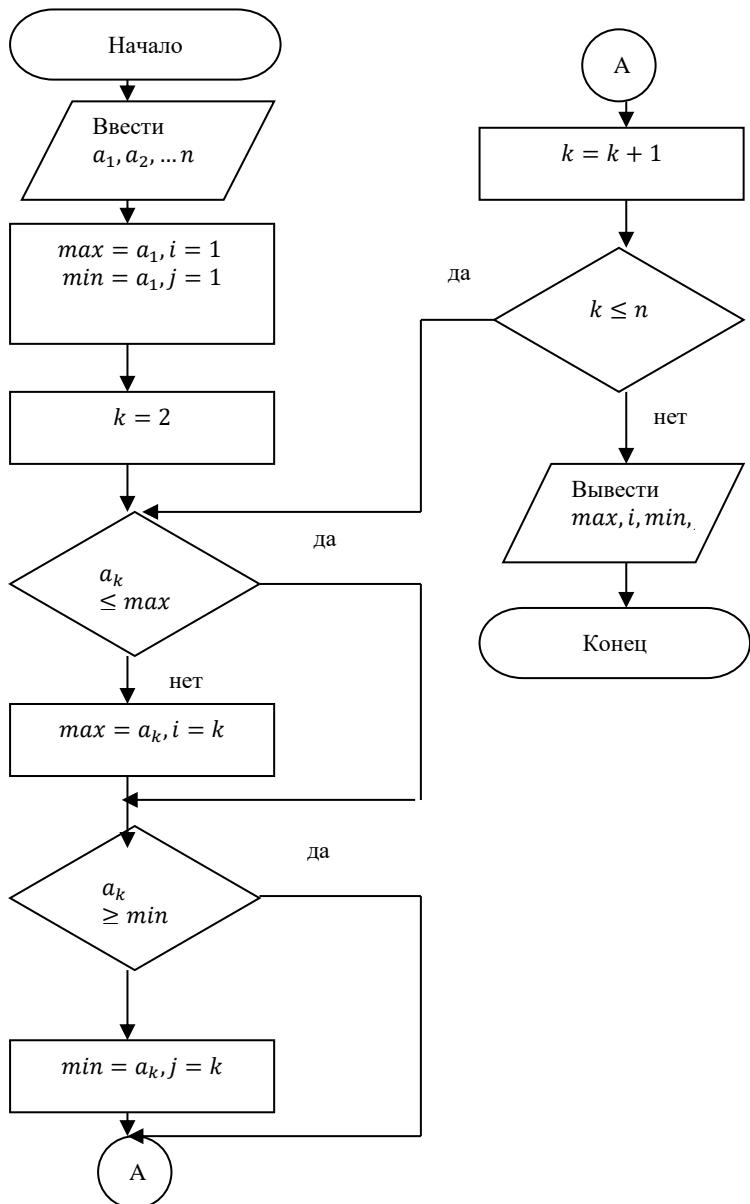


Рис. 1 Алгоритм нахождения наибольшего и наименьшего элемента в линейном массиве

Программа на языке Pascal представлена в Приложении 1, MaxMin.pas.

Бинарный поиск.

Метод бинарного поиска можно применять уже в отсортированном массиве. Допустим, что массив А отсортирован в порядке не убывания. Это позволяет по результату сравнения со средним элементом массива исключить из рассмотрения одну из половин. С оставшейся частью процедура повторяется. И так до тех пор, пока не будет найден искомый элемент или не будет построен весь массив.

Рассмотрим алгоритм бинарного поиска на примере.

Пример. Пусть $X = 6$, а массив A состоит из 10 элементов:

3 5 6 8 12 15 17 18 20 25.

1-й шаг. Найдем номер среднего элемента среди элементов: $m = \left[\frac{1+10}{2} \right] = 5$.

Так как $6 < A[5]$, то далее рассматриваются только элементы, индексы которых меньше 5.

3 5 6 8 12 15 17 18 20 25.

2-й шаг. Рассматриваем лишь первые 4 элемента массива, находим индекс среднего элемента этой части: $m = \left[\frac{1+4}{2} \right] = 2$.

$6 > A[2]$, следовательно, первый и второй элементы из рассмотрения исключаются:

~~3 5~~ 6 8 ~~12 15 17 18 20 25~~;

3-й шаг. Рассматриваем два элемента, значение $m = \left[\frac{3+4}{2} \right] = 3$.

~~3 5 6~~ 8 ~~12 15 17 18 20 25~~;

$A[3] = 6$. Элемент найден, его номер – 3.

Блок - схема алгоритма бинарного поиска на рис.2:

Программная реализация бинарного поиска представлена в Приложении 1, Binar.pas.

Случайный поиск.

Организация поиска k -го элемента в неупорядоченном массиве X возможна следующим образом. Выбирается случайным образом элемент с номером q . Массив X разбивается на три части: элементы, меньшие $X[q]$, равные $X[q]$ и большие $X[q]$. А затем, в зависимости от количества элементов в каждой части, выбирается одна из частей для дальнейшего поиска. Теоретическая оценка числа сравнений имеет порядок k^*N , т. е. для худшего случая N^2 , но на практике он значительно быстрее.

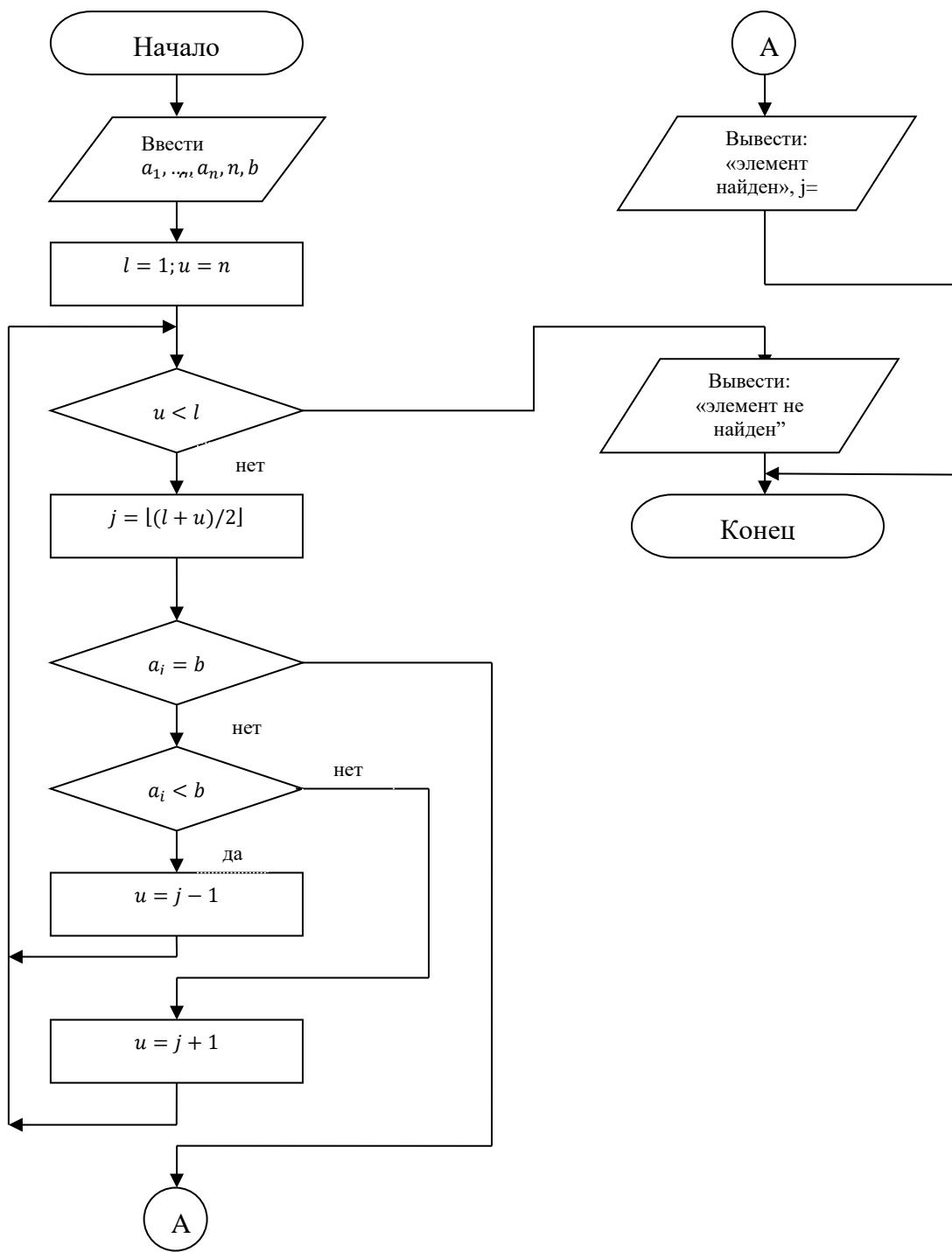


Рис. 2 Алгоритм бинарного поиска в упорядоченном массиве