

УВАЖАЕМЫЕ СТУДЕНТЫ!

ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с заданием и выполнить задания ответить на вопросы.
2. Предоставит отчет конспекта лекции прислать в виде скриншота 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

Лабораторная работа №

Тема: «Составление алгоритма поиска в неупорядоченном массиве»

Цель: изучить способы поиска элемента в неупорядоченном массиве.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Самым простым вариантом поиска можно считать поиск элемента в одномерном неупорядоченном массиве. Сформулируем задачу следующим образом: дан одномерный неупорядоченный массив, состоящий из целых чисел, и необходимо проверить, содержится ли данное число в этом массиве.

Пусть массив называется a и состоит из n элементов, а искомое число равно k . Тогда код, осуществляющий поиск, можно записать так:

```
j := -1;  
for i := 0 to n-1 do  
    if (a[i] = k) then j := 1;
```

В случае если число k ни разу не встречалось в массиве, j будет равно -1.

Приведенная выше функция будет искать последнее вхождение числа k в массиве a . Если нам необходимо искать первое вхождение, то после присваивания $j := i$ следует добавить оператор **break**; (в этом случае искомый

индекс будет храниться в переменной i).

II в том и в другом случае алгоритм будет иметь сложность $O(N)$.

На этом примере можно рассмотреть «барьерный» метод, который может быть полезен в очень многих задачах. Для использования барьерного метода наш массив должен иметь один дополнительный элемент (т.е. его длина должна быть не меньше, чем $n+1$ элемент). Отметим, что таким способом можно искать только первое вхождение элемента:

```
a[n+1] := k;  
for i := 0 to n-1 do  
    if (a[i] = k) then j := 1;
```

Если элемент k встречается в массиве, то его индекс будет находиться в переменной i , если же такой элемент в массиве не встречается, то i будет равно $n+1$.

Рассмотрим отдельно задачу поиска минимума и максимума в массиве, так же как и при поиске вхождения элемента, будем искать не само значения минимума или максимума, а индекс минимального (максимального) элемента. Это избавит нас от многих проблем и позволит совершать меньшее количество ошибок при программировании. Поиск минимального элемента в массиве, а будет выглядеть следующим образом:

```
imin := 0;  
for i := 0 to n-1 do  
    if (a[i] < a[imin]) then imin := i;
```

Индекс минимального элемента будет храниться в переменной $imin$, а сам минимум равен, $a[imin]$. Минимум и максимум следует обязательно искать по индексу, а не по значению. Например, если мы будем пытаться хранить непосредственно значение минимума или максимума, то можем легко ошибиться с начальной инициализацией. Например, для массива вещественных чисел определить значения, которыми изначально следует инициализировать минимум и максимум.

Теперь рассмотрим задачу поиска минимума и максимума одновременно. Можно реализовать такой поиск аналогично:

```
imin := 0;
imax := 0;
for i := 1 to n-1 do
  begin
    if (a[i] < a[imin]) then imin := i;
    if (a[i] > a[imax]) then imax := i;
  end;
```

Такая реализация требует $2 \times N - 2$ сравнения. Но эту задачу можно решить и за меньшее количество сравнений. Разобьем все элементы на пары, и будем искать в каждой паре минимум и максимум ($N/2$ сравнений), затем минимум будем искать только среди минимальных элементов пар, а максимум – среди максимальных. Общее количество сравнений будет около $3 \times N/2$ (проблема возникает, когда количество элементов нечетное – один из элементов остается без пары). Точно это можно записать как $\lceil 3 \times N/2 \rceil - 2$, где $\lceil \cdot \rceil$ округление до большего целого.

Рассмотрим еще один способ поиска максимума. После разбиения элементов на пары будем продолжать этот процесс, аналогично турниру «навывлет». Т.е. заново разобьем максимальные элементы из пар на пары и снова найдем максимум и т.д. Для поиска максимального элемента будет по-прежнему требовать $N - 1$ операция сравнения, но сам максимальный элемент будет участвовать только в $\log N$ сравнениях. И одно из этих сравнений обязательно будет со вторым по величине элементом. Таким образом, для поиска второго по величине элемента будет требоваться $\lceil \log N \rceil - 1$ сравнение (при условии, что все сравнения для максимального элемента проведены).

Обычно такие методы используются в особых случаях, когда это непосредственно требуется в решении задачи. Для общего случая подходит более простые методы, где количество сравнений не играет такой важной роли.

Однако и этот метод может быть полезен при поиске «порядковых статистик» массива, k -ой порядковой статистикой массива называется k -ый по счету элемент этого массива (т.е. если массив отсортировать по неубыванию, то k -ая порядковая статистика – это элемент, стоящий на k -ой позиции),

Описание массивов

В языке Паскаль массивы описываются в блоке описания переменных следующим образом:

var<идентификатор>: **array** [<тип индекса>] **of**<тип компонентов>

Здесь:

– **array** и **of** — служебные слова, которые буквально можно перевести как «массив» и «из»;

– <тип индекса>— описание индексации (нумерации) элементов массива. В качестве типа индекса можно использовать любые порядковые типы;

– <тип компонентов>— тип величин, непосредственно составляющих массив.

Приведем несколько примеров описаний:

1. **Var day:array[1..365] of integer;** –

массив, состоящий из 365 целых чисел, которые пронумерованы от 1 до 365;

2. **var tem:array[0..11] of real;** –

массив, состоящий из 12 вещественных, пронумерованных от 0 до 11;

3. **var ocenka:array[-2..2] of char;** –

массив, состоящий из 5 символьных переменных с номерами от -2 до 2:

4. **const n=10; var slovo: array[1..n] of string;** – n строковых величин, пронумерованных от 1 до n ;

Для того, чтобы обратиться к элементу массива, нужно записать имя массива и в квадратных скобках индекс нужного элемента, например, `day[100]`.

Поиск в массиве элемента, удовлетворяющего некоторому условию

Например, требуется найти в массиве элемент, значение которого равно значению переменной p , или сообщить, что такого элемента в массиве нет.

Мы построим алгоритм, идея которого следующая:

1. Просматриваем все элементы массива с первого до последнего.
2. Как только находим элемент, равный p , выведем его номер, и увеличим счетчик вхождений элемента m в массив на 1.
3. Если после просмотра массива счетчик окажется равным 0, выведем сообщение об отсутствии искомого элемента в массиве.

<pre>program massiv4; const n=10; var a: array [1..n] of integer; i, k, m, p: integer; Begin write('Введите верхнюю границу диапазона: '); readln(k); randomize; for i:=1 to n do begin a[i]:=random(k); write(a[i], ' '); end; writeln; write('Введите искомый элемент: '); readln(p); for i:=1 to n do if a[i]=p then begin writeln('Элемент ',p,' стоит в массиве под номером ',i); m:=m+1; end; if m=0 then writeln('Элемент ',p,' в массиве не найден') end.</pre>	<p>Пример работы программы:</p> <p>Введите верхнюю границу диапазона: 100 39 29 0 55 14 12 10 35 72 63 Введите искомый элемент: 10 Элемент 10 стоит в массиве под номером 7</p> <p>Введите верхнюю границу диапазона: 10 9 4 9 3 8 1 9 3 9 7 Введите искомый элемент: 9 Элемент 9 стоит в массиве под номером 1 Элемент 9 стоит в массиве под номером 3 Элемент 9 стоит в массиве под номером 7 Элемент 9 стоит в массиве под номером 9</p> <p>Введите верхнюю границу диапазона: 20 14 17 16 19 3 12 12 12 7 0 Введите искомый элемент: 10 Элемент 10 в массиве не найден</p>
---	--

Задание к лабораторной работе:

1. Изучите теоретические сведения.
2. Выполните пример, приведённый в теоретических сведениях.
3. Выполните задание согласно вашему варианту (см. табл. вариантов).
4. Подготовьте отчет, который содержит:
 - название работы, постановку цели, вывод;
 - блок-схему, текст программы и результаты ее выполнения;

ответы на контрольные вопросы, указанные преподавателем

Таблица вариантов

№ варианта	Задание
1	Найти наименьший элемент одномерного массива, состоящего из n элементов. Элементы вводятся с клавиатуры.
2	Найти номер наименьшего элемента в массиве, заданного датчиком случайных чисел на интервале $[-20, 25]$. Размер произвольный.
3	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.
4	Найти номер наибольшего элемента массива размером $M \times N$, элементы которого задаются датчиком случайных чисел на интервале $[-67; 23]$.
5	Найти номер наибольшего элемента массива размером $M \times N$, элементы которого вводятся с клавиатуры.
6	В двумерном массиве, состоящем из целых чисел, найти наименьший элемент и номер строки, в которой он находится. Элементы вводятся с клавиатуры. Размер $M \times N$.
7	Массив A вводится с клавиатуры. Вывести только нечетные элементы. Размер произвольный.
8	Найти наибольший элемент и его номер в последовательности, элементы которой вводятся с клавиатуры. Размер произвольный.
9	Найдите среднее арифметическое элементов массива, состоящего из 10 чисел, которые превышают по величине число C . Элементы вводятся с клавиатуры.
10	Найти наименьший элемент двумерного массива. Размер $M \times N$. Элементы задаются на интервале $[-30, 45]$.
11	Найти наибольший элемент и его номер в последовательности, элементы которой вводятся с клавиатуры. Размер произвольный.
12	В двумерном массиве, состоящем из целых чисел, найти наибольший элемент и номер строки, в которой он находится. Элементы вводятся с клавиатуры. Размер $M \times N$.
13	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.

14	Найти номер наименьшего элемента в массиве, заданного датчиком случайных чисел на интервале $[-30, 25]$. Размер произвольный.
15	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.
16	Найти номер наименьшего элемента массива размером $M \times N$, элементы которого вводятся с клавиатуры.
17	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Для ввода элементов использовать функцию <code>random</code> .
18	Найти наименьший элемент одномерного массива, состоящего из n элементов. Элементы вводятся с клавиатуры.
19	Найти номер наименьшего элемента в массиве, заданного датчиком случайных чисел на интервале $[-50, 25]$. Размер произвольный.
20	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.
21	Найти наименьший элемент двумерного массива. Размер $M \times N$. Элементы задаются на интервале $[-20, 35]$.
22	Найти номер столбца массива размером $M \times N$, в котором находится наибольшее количество отрицательных элементов. Элементы вводятся с клавиатуры.
23	Найти номер наибольшего элемента массива размером $M \times N$, элементы которого задаются датчиком случайных чисел на интервале $[-7; 23]$.
24	Найти номер наибольшего элемента массива размером $M \times N$, элементы которого вводятся с клавиатуры.
25	В двумерном массиве, состоящем из целых чисел, найти наименьший элемент и номер строки, в которой он находится. Элементы вводятся с клавиатуры. Размер $M \times N$.

Контрольные вопросы:

1. Что такое массив?
2. Что такое идентификатор массива?
3. Что такое переменная с индексом?
4. Какие значения принимает индекс у переменных массива на языке

Паскаль?

5. Верно ли, что массив – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип?

6. Какие типы данных недопустимы для компонентов массива? Почему?

7. В каком разделе (разделах) программы допустимо описание данных регулярного типа?

8. Где и как определяется общее число элементов массива?

9. Можно ли размерность массива определить с помощью типа диапазонов?

10. Дайте определение индекса. Какие типы данных можно использовать при описании индекса?