

## 22 апреля

### Классная работа

#### Тема: Решение задач на обработку массивов.

Сортировкой или упорядочиванием массива называется расположение его элементов по возрастанию (или убыванию).

В общем, это сложная тема, в которой известно много различных алгоритмов. Критерии оценки эффективности этих алгоритмов могут включать следующие параметры:

- количество шагов алгоритма, необходимых для упорядочения;
- количество сравнений элементов;
- количество перестановок, выполняемых при сортировке.

Мы рассмотрим только два простейших метода сортировки Метод «пузырька» и Метод поиска минимума.

#### 1. Метод «пузырька»

Самым простым методом сортировки является так называемый метод «Пузырька». Чтобы понять его идею, представьте, что массив расположен вертикально. Элементы с большим значением всплывают вверх подобно большому пузырькам. При первом проходе вдоль массива, начиная проход «снизу», берется первый элемент и по очереди сравнивается с последующим. При этом:

- если встречается больше «легкий» (с меньшим значением) элемент, то они меняются местами;
- при встрече с более «тяжелым» элементом, последний становится «эталоном» для сравнения, и все последующие сравниваются с ним.

В результате наибольший элемент оказывается в самом верху массива.

Во время второго прохода вдоль массива находится второй по величине элемент, содержащийся под элементом, найденным при первом проходе, т.е. на вторую сверху позицию, и т.д.

Заметим, что при втором и последующих проходах, нет необходимости рассматривать ранее «всплывшие» элементы, потому что они очевидно больше оставшихся. Иначе говоря, при  $j$ -ом проходе не проверяются элементы, стоящие на позициях выше  $j$ .

Теперь можно привести текст программы упорядочения массива  $M [1..N]$ :

```
for j: = 1 to N-1 do
  for i: = 1 to N-j do
    if mass[i] > mass[i + 1] then
      begin
        min := mass[i];
        mass[i] := mass[i + 1];
        mass[i + 1] := min;
      end;
```

#### 2. Сортировка поиском минимума

На каждом шаге наш алгоритм находит наименьшее число в исходном наборе, удаляет его оттуда и записывает в конец набора, представляющий результат. Такой способ называют сортировкой поиском минимума.

Пусть нам дан массив из  $N$  чисел. Заметим, что количество чисел, которые мы сохраняем, по ходу работы алгоритма не меняется: на каждом шаге мы просто переносим одно из чисел из исходного набора в результирующий. Поэтому нам нет нужды заводить новый массив для хранения результата: мы сможем хранить его в освободившихся ячейках старого массива.

Разберем подробнее, как это можно реализовать. Для этого снова обратимся к нашему примеру: 3, 6, 1, 2, 9, 5.

- На первом этапе мы находим наименьшее число - 1 должны переместить на первое место. Но там уже записано число 3, куда же его деть. Ответ прост: на место числа 1. Таким образом, мы просто меняем местами два элемента массива: 1 | 6, 3, 2, 9, 5.
- На втором шаге мы находим минимальное число в правой части - среди элементов массива с 2-го по 6-й (в общем случае - по  $N$ -й) - и меняем его местами с числом, которое стоит на втором месте: 1, 2 | 3, 6, 9, 5.

- Далее нам следует задать следующее минимальное число - 3 - на третье место, но оно уже и так там стоит. Впрочем, мы не будем обращать на это внимание, и просто как бы поменяем его именно с собой: 1, 2, 3 | 6, 9, 5.
- Попробуем записать действия, которые нам предстоит проделать на этапе с номером k. Сначала нам нужно найти наименьшее среди чисел, записанных в k, k + 1, ..., N позициях массива. После этого нам нужно поменять этот элемент из k-м.

Сколько таких шагов нужно сделать, чтобы полностью отсортировать массив? Напрашивается ответ "N", но не будем торопиться. Пусть сделан N-1 шаг. Тогда в правой, неотсортированной части массива останется одно число. Какое это число? Ясно, что это наибольшее число в массиве. Где оно в итоге должно оказаться? На последнем месте в массиве. Но оно уже и так там! Поэтому N-й шаг алгоритма выполнять не нужно.

Таким образом, алгоритм можно записать следующим образом:

Будем считать, что массив называется mass, и напомним соответствующий код, вспомнив, как искать минимальный элемент и как менять два элемента местами:

```
for i: = 1 to N-1 do
  begin
    min = i;
    for j = i + 1 to N do
      if mass[j] < mass[min] then min = j;
    temp := mass[min];
    mass[min] := mass[i];
    mass[j] := temp;
  end;
```

Алгоритм сортировки поиском минимума реализован.

### Пример:

- Заполнить массив 20-ю случайными целыми числами из промежутка [-5;20] и вывести на экран отсортированный массив.

```
Program sortirovka;
var mass:array[1..20] of integer;
    i,j,max:integer;
Begin
  randomize;
  for i:=1 to 20 do
    begin
      mass[i]:=random(26)-5;
      write(mass[i], ' ');
    end;
  writeln;
  for j:=1 to 19 do
    for i:=1 to 20-j do
      if mass[i]>mass[i+1] then
        begin
          max:=mass[i];
          mass[i]:=mass[i+1];
          mass[i+1]:=max;
        end;
    end;
  for i:=1 to 20 do write(mass[i], ' ');
end.
```

} Заполнение массива  
 } Сортировка массива  
 } Вывод массива