Сортировка элементов линейного массива

Сортировка элементов массива

Сортировка – один из наиболее распространенных процессов обработки данных.

Сортировкой числового массива называют расположение его элементов в возрастающем или убывающем по величине порядке.

Сортировка элементов массива

Под *сортировкой массива* подразумевается процесс перестановки элементов с целью упорядочивания их в соответствии с каким-либо критерием.

Существует достаточно много методов (алгоритмов) сортировки массивов. Рассмотрим *метод прямого выбора* и *метод парных перестановок.*

Метод прямого выбора

Алгоритм сортировки массива по возрастанию методом прямого выбора может быть представлен так:

- *Просматривая массив с первого элемента, найти минимальный и поменять его местами с первым элементом.*
- *Просматривая массив со второго элемента, найти минимальный и поменять его местами со вторым элементом.*
- *И, так далее, до последнего элемента.*

Метод прямого выбора

Алгоритм использует вложенные циклы.

Внешний цикл (счетчик шагов) последовательно выбирает номер элемента массива, куда следует записывать найденный в неупорядоченной части массива минимальный элемент.

Внутренний цикл перебирает номера неупорядоченных элементов при поиске минимального элемента.

Для внешнего цикла достаточно шагов на один меньше, чем элементов в массиве.

Метод прямого выбора

Фрагмент программы, реализующий сортировку массива по возрастанию методом прямого выбора

```
for i:= 1 to n-1 do
    for j:= i+1 to n do
        if (A[j]<A[i]) then
            swap(A[j], A[i]);
```

Метод прямого выбора

Фрагмент программы, реализующий сортировку массива по возрастанию методом прямого выбора

```
for i:= 1 to n-1 do
    for j:= i+1 to n do
        if (A[j]<A[i]) then
            begin
                z:=A[i];
                A[i]:=A[j];
                A[j]:=z;
            end;
```

```
program s_1;
var a: array [1..10000] of integer;
var i, j, N, z: integer;
begin
   cls;
   write('Количество элементов массива ');
   read(N);
   randomize;
   for i:=1 to N do
      begin
         A[i]:=random(100)-50;
         write(A[i]:5);
      end;
   writeln();
   for i:= 1 to n-1 do
      for j:= i+1 to n do
         if (A[j]<A[i]) then
         begin
            z:=A[i];
            A[i]:=A[j];
            A[j]:=z;
```

Метод прямого выбора

Пример работы алгоритма:

Исходный массив: 8, 3, 6, 1, 4 (последовательно меняются местами 8 и 3, 3 и 1)

После первого шага: **1**, 8, 6, 3, 4 (меняются местами 8 и 6, 6 и 3)

После второго шага: **1**, **3**, 8, 6, 4 (меняются местами 8 и 6, 6 и 4)

После третьего шага: **1**, **3**, **4**, 8, 6 (меняются местами 8 и 6)

После четвертого шага: **1**, **3**, **4**, **6**, **8**

Метод парных перестановок (пузырьковый)

Смысл этого метода заключается в сравнивании соседних элементов и, если нужно, их перестановке. Причём за один просмотр всех пар сортировка не достигает нужного результата. Приходится просматривать все пары элементов несколько раз.

Если мы сортируем массив по возрастанию, то суть метода «пузырька» будет сводится к тому, что постепенно самые большие элементы будут "оседать" в конце массива, а меньшие - постепенно "всплывать" к его началу (как пузырьки воздуха в воде).

Метод парных перестановок (пузырьковый)

Алгоритм сортировки включает два цикла. Один вложен в другой. За каждый повторений внешнего цикла самый большой элемент в просматриваемом отрезке массива устанавливается в конец этого отрезка.

Во внутреннем цикле сравниваются соседние элементы. Если очередной больше следующего (при сортировке по возрастанию), то происходит их обмен.

Метод парных перестановок (пузырьковый)

Количество повторений внутреннего цикла равно количеству элементов в массиве минус номер повторения внешнего. Так регулируется длина отрезка массива, который необходимо просматривать. Счетчик внутреннего цикла - это индекс элемента массива.

Метод парных перестановок (пузырьковый)

Фрагмент программы, реализующий сортировку массива по возрастанию методом «пузырька»

```
for j:= 1 to n-1 do
    for i:= 1 to n-j do
        if (A[i]>A[i+1]) then
            swap(A[i], A[i+1]);
```

Метод парных перестановок (пузырьковый)

Фрагмент программы, реализующий сортировку массива по возрастанию методом «пузырька»

```
for j:= 1 to n-1 do
   for i:= 1 to n-j do
      if (A[i]>A[i+1]) then
         begin
            z:=A[i];
            A[i]:=A[i+1];
            A[i+1]:=z;
         end;
```

```
program s_1;
var a: array [1..10000] of integer;
var i, j, N, z: integer;
begin
   cls;
   write('Количество элементов массива ');
   read(N);
   randomize;
   for i:=1 to N do
      begin
         A[i]:=random(100)-50;
         write(A[i]:5);
      end;
   writeln();
   for j:= 1 to n-1 do
      for i:= 1 to n-j do
         if (A[i]>A[i+1]) then
         begin
            z:=A[i];
            A[i]:=A[j];
            A[j]:=z;
         end;
```

Метод парных перестановок (пузырьковый)

1 3 5 2 4

Сравниваем первую пару элементов. Числа 1 и 3 стоят в правильном порядке. Их переставлять не надо. Делаем один шаг по массиву и сравниваем числа 3 и 5. Их тоже не надо переставлять. Делаем еще один шаг и сравниваем числа 5 и 2. Они стоят в неправильном порядке, поэтому мы их меняем местами.

Метод парных перестановок (пузырьковый)

1 3 2 5 4

Сдвигаемся еще на один элемент и сравниваем числа 5 и 4. Они тоже стоят неправильно, и мы меняем их местами.

1 3 2 4 5

Мы дошли до конца массива. Но массив пока неупорядочен. Поэтому мы возвращаемся к началу массива и продолжаем сравнение. Числа 1 и 3 идут в правильном порядке, а вот числа 3 и 2 порядок нарушают, поэтому мы меняем их местами.

Метод парных перестановок (пузырьковый)

1 2 3 4 5

Пройдя еще один раз по всему массиву, мы убеждаемся, что все элементы стоят на своих местах. Значит, процесс сортировки массива можно завершить.

Задачи

1. Составить программу сортировки числового массива по убыванию методом прямого выбора. Массив задать случайными числами в интервале от 0 до 100.

2. На соревнованиях по прыжкам в длину получен массив b(n). Определить три лучших результата. Массив сформировать с помощью функции random (диапазон прыжков от 120см до 200см).

Задачи

3. Составить программу сортировки числового массива по убыванию так, чтобы осуществлялась сортировка:

1) четных элементов массива;

2) отрицательных элементов массива;

3) элементов, записанных на нечетных местах.

4. Определить есть ли в массиве равные элементы. Вывести их на экран.