Алгоритмы сортировки массивов

Для решения многих задач удобно сначала упорядочить данные по определенному признаку, так можно ускорить поиск некоторого объекта. Например, возьмем любой энциклопедический словарь – статьи в нем упорядочены в алфавитном порядке.

Перегруппирование заданного множества объектов в определенном порядке называют ***сортировкой*.**

"Даже если бы сортировка была почти бесполезна, нашлась бы масса причин заняться ею! Изобретательные методы сортировки говорят о том, что она и сама по себе интересна как объект исследования."

/Д. Кнут/

"Создается впечатление, что можно построить целый курс программирования, выбирая примеры только из задач сортировки."

/Н. Вирт/

Отличительной особенностью сортировки является то обстоятельство, что эффективность алгоритмов, реализующих ее, прямо пропорциональна сложности понимания этого алгоритма. Другими словами, чем легче для понимания метод сортировки массива, тем ниже его эффективность.

При решении задачи сортировки обычно выдвигается требование минимального использования дополнительной памяти, из которого вытекает недопустимость применения дополнительных массивов.

Для оценки быстродействия алгоритмов различных методов сортировки, как правило, используют два показателя:

- количество присваиваний;
- количество сравнений.

Все методы сортировки можно разделить на две большие группы:

- прямые методы сортировки;
- улучшенные методы сортировки.

Прямые методы сортировки по принципу, лежащему в основе метода, в свою очередь разделяются на три подгруппы:

1) сортировка вставкой (включением);
2) сортировка выбором (выделением);
3) сортировка обменом (так называемая "пузырьковая"сортировка).

Улучшенные методы сортировки основываются на тех же принципах, что и прямые, но используют некоторые оригинальные идеи для ускорения процесса.

Сортировка вставкой.

Принцип метода заключается в следующем:

массив разделяется на две части: отсортированную и не отсортированную. элементы из не отсортированной части поочередно выбираются и вставляются в отсортированную часть так, чтобы не нарушить в ней упорядоченность элементов. В начале работы алгоритма в качестве отсортированной части массива принимают только первый элемент, а в качестве не отсортированной – все остальные элементы.

Таким образом, алгоритм будет состоять из (n-1)-го прохода (n – размерность массива), каждый из которых будет включать четыре действия:

- взятие очередного i-го не отсортированного элемента и сохранение его в дополнительной переменной;
- поиск позиции j в отсортированной части массива, в которой присутствие взятого элемента не нарушит упорядоченности элементов;
- сдвиг элементов массива от i-го до j-1-го вправо, чтобы освободить найденную позицию вставки;
- вставка взятого элемента в найденную i-ю позицию.

Задание: Напишите процедуру, реализующую выше рассмотренный алгоритм:

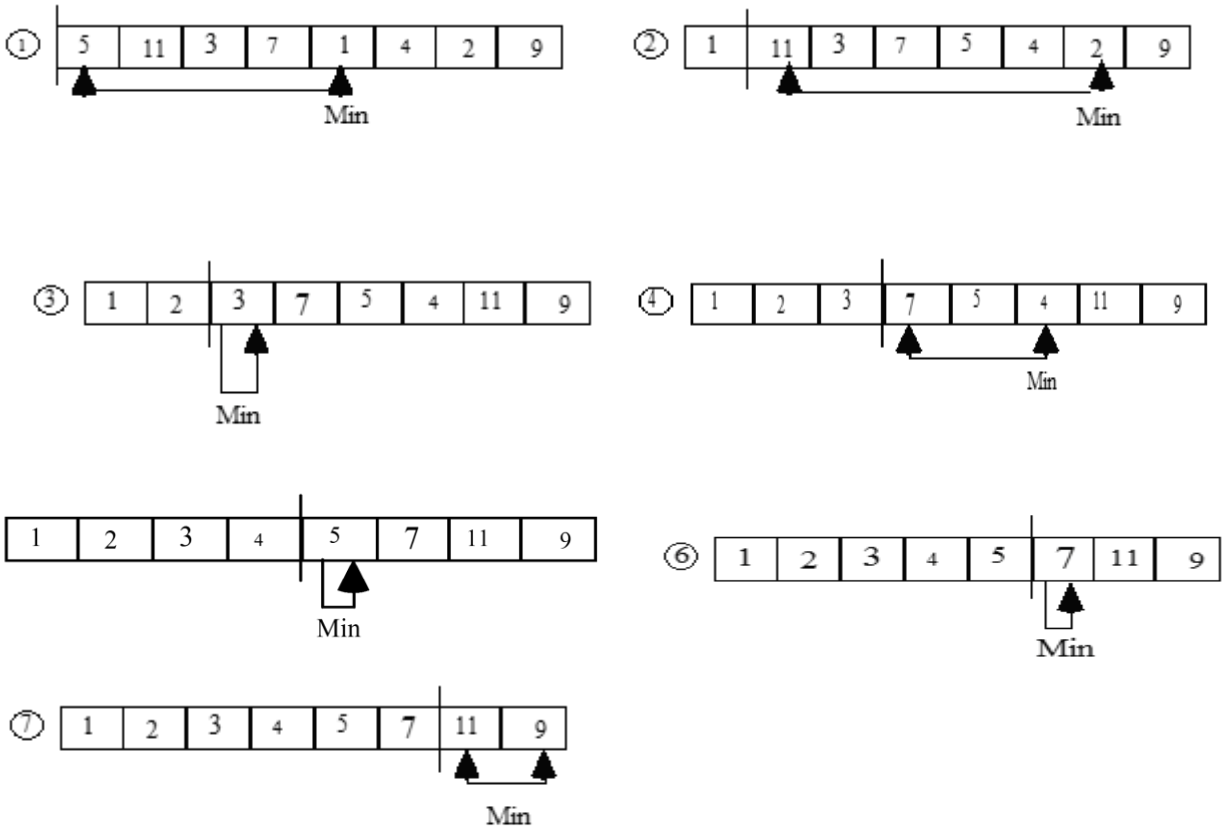
Пример процедуры

```
Procedure Vstavka(Var a : mas1);
Var
  i, j,e,g:Integer;
Begin
  For i:=2 to c do
    Begin
      e:=A[i];
      j:=1;
      While (e>a[j]) do
         Inc(j);
      For g:=i-1 downto j do
        a[g+1]:=a[g];
      a[j]:=e;
    End;
End;
```

Сортировка выбором.

Принцип метода:
Находим (выбираем) в массиве элемент с минимальным значением на интервале от 1-го элемента до n-го (последнего) элемента и меняем его местами с первым элементом. На втором шаге находим элемент с минимальным значением на интервале от 2-го до n-го элемента и меняем его местами со вторым элементом. И так далее для всех элементов до n-1-го.

Рассмотрите схему алгоритма прямого выбора.



Задание: Напишите процедуру, реализующую выше рассмотренный алгоритм:

Пример процедуры

```
Procedure Vibor(Var a: mas1);
Var
  i, j, Min, MinI : Integer;
Begin
  For i:=1 to c do
    Begin
      Min:=a[i];
      MinI:=i;
      For j:=i+1 to c do
        If a[j]<Min
          Then
            Begin
              Min:=a[j];
              MinI:=j;
            End;
      a[MinI]:=a[i];
               a[i]:=Min;
    end;
End;
```

Сортировка методом простого обмена.

Принцип метода:

Слева направо поочередно сравниваются два соседних элемента, и если их взаимное расположение не соответствует заданному условию упорядоченности, то они меняются местами. Далее берутся два следующих соседних элемента и так далее до конца массива.

После одного такого прохода на последней n-ой позиции массива будет стоять максимальный (или минимальный) элемент ("всплыл" первый "пузырек"). Поскольку максимальный (или минимальный) элемент уже стоит на своей последней позиции, то второй проход обменов выполняется до (n-1)-го элемента. И так далее.

Всего требуется (n-1) проход.

Задание: Напишите процедуру, реализующую выше рассмотренный алгоритм:

Пример процедуры

```
Procedure Obmen(Var a : mas1);
Var
  i,j,f,g:Integer;
Begin
  For i:=n downto 2 do
    For j:=1 to i-1 do
      If a[j]>a[j+1]
        Then
          Begin
             f:=a[j];
            a[j]:=a[j+1];
            a[j+1]:=f;
          End;
End;
```

Другие методы (предлагаются сильным ученикам):

- Сортировка методом слияний.
- Шейкерная сортировка.
- Гномья сортировка
- Быстрая сортировка.

Список электронных ресурсов:
http://kpolyakov.spb.ru/
http://wwwfoxford.ru
http://informatics.mccme.ru/
http://www.infourok.ru
http://edunow.su/site/content/algorithms/sortirovka_massiva наглядное представление алгоритмов сортировки (видео – танцы)