Муниципальное бюджетное учреждение дополнительного образования
«Городской центр детского технического творчества
им. В.П.Чкалова» г. Казани

16 городской Конкурс-выставка технического творчества школьников «Дети. Техника. Творчество»

Сортировка целочисленного массива методов вставок

Рекомендации для педагогов дополнительного образования по программированию

Потопахин Виталий Валерьевич,
педагог дополнительного образования

Сортировка массивов

На этом уроке, мы разберем классическую программисткую задачу сортировки целочисленного массива. Результатом работы будет решение задачи на псевдокоде, которое затем перепишем на языке программирования - Компонентный Паскаль.

Постановка задачи

Общую задачу сортировки можно определить, как перестановка элементов массива в некотором заданном порядке. Мы, далее будем рассматривать простую и самую распространенную задачу сортировку по возрастанию. Ее смысл в том, что алгоритм на вход принимает произвольный массив, а на выход выдает массив содержащий те же элементы, но упорядоченные в порядке возрастания.

Пример сортировки

Пусть например на вход дан следующий массив: 4, 1, 3, 89, 4, 232

Тогда массив результат должен выглядеть так: 1, 3, 4, 4, 89, 232

Метод сортировки вставками

Суть метода заключается в следующем – для каждого элемента массива определяется его «правильная позиция». То есть такая, что левее его все элементы массива будут меньше. В примере ниже для числа выделенного красным, зеленым обозначена правильная позиция:

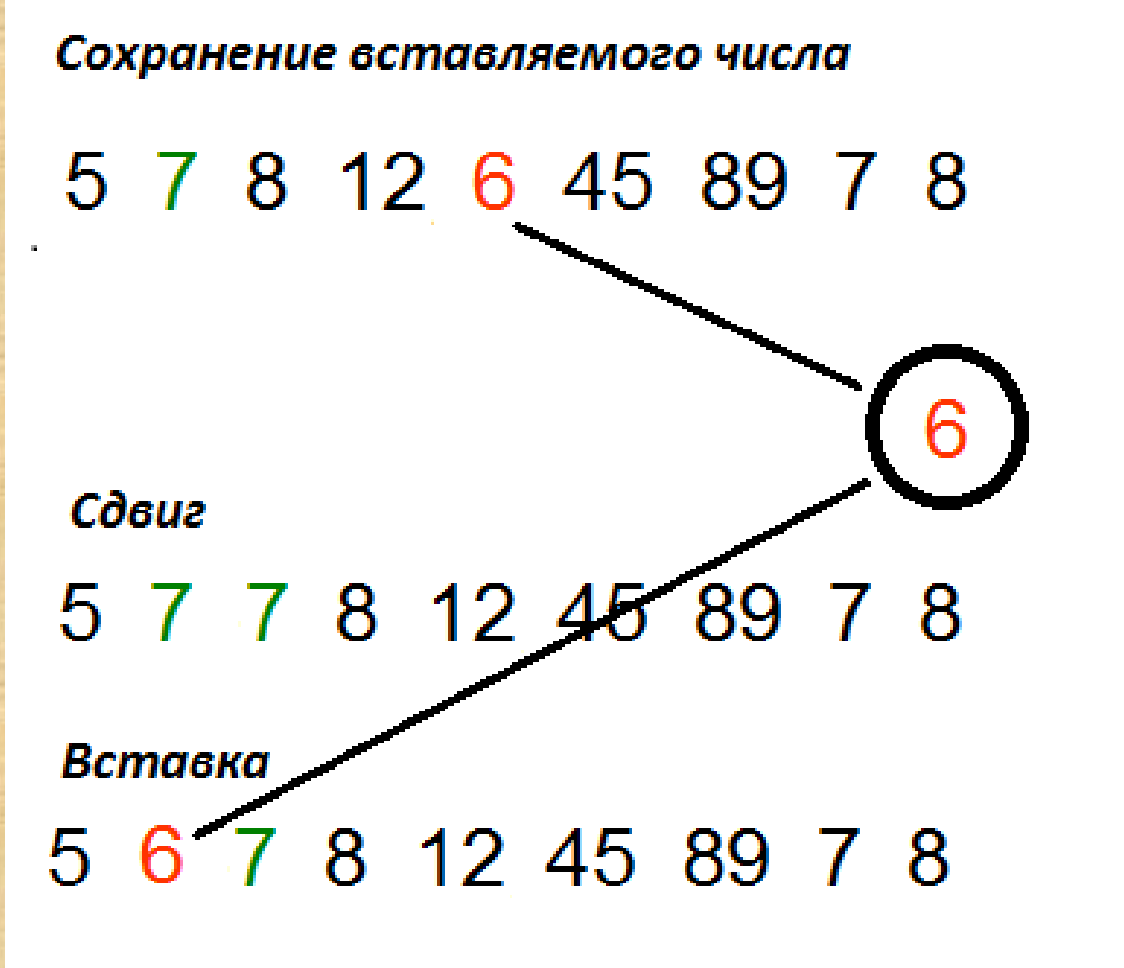
5, 7, 8, 12, 6, 45, 89, 7, 8

Метод сортировки вставками

5, 7, 8, 12, 6, 45, 89, 7, 8

Чтобы вставить число 6 в позицию числа 7 необходимо часть массива от 7 до 6 (включительно) сдвинуть на одну позицию вправо, а затем число 6 вставить вместо 7. Перед сдвигом число 6 потребуется сохранить в специальной переменной.

Сортировка методом вставок



Алгоритм метода вставок

Любая работа с массивом начинается с его ввода с клавиатуры или как иногда говорят из потока. Ниже запись стандартной процедуры:

Ввести N (Количество чисел)

Для k от 0 до N-1 делать

    Ввести элемент A(k)

Конец ввода

Здесь записан так называемый цикл – многократно выполняемая команда ввода. На каждом ее шаге вводится новый элемент массива с номерами от 0 до N-1

Алгоритм метода вставок

На необходимо принять решение о вставке для каждого элемента массива за исключением стоящего в нулевой позиции. Левее нулевого нет ничего, поэтому он в начале процесса стоит в правильной позиции. Но элементы с первого по последний необходимо проанализировать, для чего запишем циклическую конструкцию:

Для k от 1 до N-1 делать

Выполнить анализ и если потребуется вставку элемента массива с номером k

Конец обработки

Алгоритм метода вставок

Анализ заключается в поиске левее числа уже большего, нежели анализируемое, вставка потребуется только в том случае, если таковое будет найдено

```
Для k от 1 до N-1 делать
    j=0
    Пока A(j)<A(k) Делать
        j=j+1
    Конец поиска
    Если j<k То
        Выполнить вставку
Конец обработки
```

Если левее будет найдено меньшее, то очевидно его номер j окажется меньше номера анализируемого числа k, что и станет сигналом к необходимости вставки.

Алгоритм метода вставок

Вставка включается в себя три операции: сохранение вставляемого числа. Сдвиг массива от найденного до сохраняемого и вставку сохраняемого на место найденного. Сказанное можно записать так:

```
Для k от 1 до N-1 делать
      j=0
     Пока A(j)<A(k) Делать
          j=j+1
    Конец поиска
    Если j<k То
         c=A(k)
         Сдвиг
         A(j)=c
  Конец обработки
```

Осталось отработать процедуру сдвига.

Сдвиг массива

Это наиболее сложная операция, поэтому опишем ее отдельно. Суть операции сдвига в том, что элемент с номером k-1, необходимо поставить в позицию k, элемент с номером k-2, в позицию k-1, и так далее до элемента с номером j. Это можно записать так:

t=k

Пока t>j делать

A(t)=A(t-1) Это собственно сдвиг

t=t-1 Переход к следующему элементу для сдвига

Конец сдвига

Полная запись алгоритма

```
Ввести N
Для k от 0 до N-1 делать
   Ввести элемент A(k)
Конец ввода
Для k от 1 до N-1 делать
        j=0
       Пока A(j)<A(k) Делать
            j=j+1
      Конец поиска
      Если j<k То
          c=A(k)
          t=k
        Пока t>j делать
             A(t)=A(t-1)
             t=t-1
        Конец сдвига
          A(j)=c
Конец главного цикла обработки
Для k от 0 до N-1 делать
   Напечатать элемент A(k)
Конец вывода
```

Разработка программы

Начнем разработку с описания общей

структуры программы. На языке Компонентный паскаль, для этого необходимо создать модуль, подключить как минимум библиотеки ввода-вывода и описать процедуру, содержащую решение задачи.

```
MODULE ЗадачиСортировки;
    IMPORT In, StdLog;
PROCEDURE СортировкаВставками;
BEGIN
END СортировкаВставками;
END ЗадачиСортировки.
```

Разработка программы

Следующим шагом опишем ввода массива и объявим необходимые для работы программы переменные

```
MODULE ЗадачиСортировки;
    IMPORT In, StdLog;
PROCEDURE СортировкаВставками;
   VAR
      a:ARRAY 100 OF INTEGER;
      k,k,t,c:INTEGER;
BEGIN
    In.Open;
    In.Int(N);
    FOR i:=0 TO N-1 DO
       In.Int(a[i]);
    END;
    (*Собственно алгоритм сортировки*)
END СортировкаВставками;
END ЗадачиСортировки.
```

Разработка программы

Сейчас займемся логической частью программы. Ее главный цикл запишем с помощью цикла по параметру, такого же как мы использовали для ввода данных, с учетом того обстоятельства, что обработка начинается не с нулевого, а первого элемента.

```
FOR k:=1 TO N-1 DO
   Анализ элемента с номером k
END;
```

Разработка программы

В алгоритме анализ заключается в поиске левее k-го элемента, такого элемента с номером j, который уже больше чем k-ый. Этот фрагмент запишется так:

```
FOR k:=1 TO N-1 DO
  j:=0;
  WHILE a[j]<a[k] DO j:=j+1; END;
  IF j<k THEN
     Сохранение k-го элемента
     Сдвиг фрагмента массива
     Вставка k-го элемента в позиции j
  END;
END;
```

Разработка программы

И наконец последние команды алгоритма

```
FOR k:=1 TO N-1 DO
      j:=0;
      WHILE a[j]<a[k] DO j:=j+1; END;
      IF j<k THEN
        c:=a[k];
        t:=k;
        WHILE t>j DO
          a[t]:=a[t-1];
          t:=t-1;
        END;
        a[j]:=c;
      END;
    END;
```

Разработка программы

И наконец совсем последнее действие это операторы вывода:

```
FOR k:=0 TO N-1 DO
   StdLog.Int(a[kl]);
END;
```

Полный текст прогарммы

```
MODULE ЗадачиСортировки;
    IMPORT In, StdLog;
PROCEDURE СортировкаВставками;
   VAR
      a:ARRAY 100 OF INTEGER;
      k,k,t,c:INTEGER;
BEGIN
    In.Open; In.Int(N);
    FOR i:=0 TO N-1 DO In.Int(a[i]); END;
    FOR k:=1 TO N-1 DO
        j:=0;
       WHILE a[j]<a[k] DO j:=j+1; END;
          IF j<k THEN
             c:=a[k]; t:=k;
             WHILE t>j DO a[t]:=a[t-1]; t:=t-1; END;
             a[j]:=c;
          END;
      END;
FOR k:=0 TO N-1 DO  StdLog.Int(a[kl]); END;
END СортировкаВставками;
END ЗадачиСортировки.
```