

Лекции 8, 9.

Некоторые приёмы программирования

Наиболее часто используемые приёмы программирования, применяемые студентами на практических занятиях, рассматриваются на примерах обработки числового массива A с количеством элементов N . Ниже представлены фрагменты программ на языке **Pascal** для каждого из предложенных приемов.

Вычисление суммы и произведения элементов массива:

```

.....
S:=0;                      // сумма
P:=1;                      // произведение
FOR I:=1 TO N DO
  BEGIN
    S:= S+A[I];
    P:= P*A[I];
  END;
.....

```

Нахождение максимального (минимального) элемента массива (на примере нахождения максимального элемента):

```

.....
IMAX:=1;
FOR I:=2 TO N DO
  IF A[I]>A[IMAX]
  THEN
    IMAX:=I;
.....

```

Здесь $IMAX$ – номер максимального элемента;
 $A[IMAX]$ – максимальный элемент.

Удаление заданного (K -го) элемента массива:

```

.....
FOR I:=K TO N-1 DO
    A[I]:=A[I+1];
N:=N-1;
.....

```

Вставка нового элемента (M) на заданное (K -е) место в массив:

```

.....
N:=N+1;
FOR I:=N DOWNT0 K+1 DO
    A[I]:=A[I-1];
A[K]:=M;
.....

```

Сортировка массива**1. Метод обмена («метод пузырька»):**

Суть данного метода заключается в сравнении соседних элементов массива, начиная с первой и до последней пары. Если предыдущий элемент пары больше последующего (при сортировке по возрастанию), то соседи меняются местами. Для полной сортировки необходимо сделать $N-1$ таких «проходов» по массиву.

```

.....
FOR J:= 1 TO N-1 DO
    FOR I:= 1 TO N-1 DO
        IF A[I] > A[I+1]
            THEN
                BEGIN
                    BUF:= A[I];
                    A[I]:= A[I+1];
                    A[I+1]:= BUF;
                END;
.....

```

Здесь *BUF* – буферная переменная для перестановки;

Метод обмена можно ускорить двумя способами:

а) уменьшать на каждом проходе количество сравниваемых пар:

```

.....
FOR J:= 1 TO N-1 DO      // или  FOR J:=N-1 DOWNT0 1 DO
FOR I:= 1 TO N-J DO      //      FOR I:=1 TO J DO
  IF A[I] > A[I+1]
  THEN
    BEGIN
      BUF:= A[I];
      A[I]:= A[I+1];
      A[I+1]:= BUF;
    END;
.....

```

б) прекращать сортировку при окончании перестановок:

```

.....
K:=N;
REPEAT
  PORYADOK:=TRUE;
  K:= K-1;
  FOR I:= 1 TO K DO
    IF A[I] > A[I+1]
    THEN
      BEGIN
        BUF:= A[I];
        A[I]:= A[I+1];
        A[I+1]:= BUF;
        PORYADOK:=FALSE;
      END;
  UNTIL PORYADOK;
.....

```

2. Метод выбора:

Суть метода выбора: при сортировке по возрастанию находится минимальный элемент в диапазоне от первого до последнего и меняется местами с первым. Далее находится минимальный элемент от второго до последнего и меняется со вторым и т.д.

```

.....
FOR I:= 1 TO N-1 DO
BEGIN
  MIN:= A[I];
  IMIN:=I;
  FOR J:= I+1 TO N DO
    IF A[J] < MIN
      THEN
        BEGIN
          MIN:= A[J];
          IMIN:= J;
        END;
  A[IMIN]:= A[I];
  A[I]:= MIN;
END;
.....

```

3. Метод вставки:

Считается, что массив разделён на две части: отсортированную (в начале массива) и неотсортированную (в остальной части массива). В самом начале отсортированная часть состоит из первого элемента. Первый элемент из неотсортированной части (его номер – I) вставляется в отсортированную часть без изменения порядка (в позицию J):

```

.....
FOR I:= 2 TO N DO
BEGIN
  BUF:= A[I];
  J:= 1;
  WHILE BUF > A[J] DO
    J:= J+1;
  FOR K:= I DOWNT0 J+1 DO
    A[K]:= A[K-1];
  A[J]:=BUF;
END;
.....

```

Поиск в массиве

1. Метод перебора:

Просматриваются элементы массива, начиная с первого, и сравниваются с искомым значением до тех пор, пока не произойдёт совпадение или не будет просмотрен весь массив.

```

.....
I:= 0;
NAIDEN:= FALSE;
REPEAT
  I:= I+1;
  IF A[I] = ISKOMOE
    THEN
      NAIDEN:=TRUE;
UNTIL NAIDEN OR (I=N);
IF NAIDEN
  THEN
    WRITELN('Элемент найден, его номер - ',I)
  ELSE
    WRITELN('Элемент не найден');
.....

```

2. Метод бинарного поиска:

Метод бинарного поиска (метод деления пополам) используется только для упорядоченных массивов. Суть его заключается в том, что находится центральный (серединный) элемент массива и сравнивается с искомым. Если они равны, то поиск прекращается. Если они не равны, то, если искомый элемент больше центрального (при сортировке по возрастанию), из рассмотрения исключается половина массива от первого до центрального элемента включительно. Если же искомый элемент меньше центрального, то исключается часть массива, начиная от центрального до последнего элемента. В остальной части находится центральный элемент и сравнивается с искомым и т.д. до тех пор, как произойдёт совпадение или начало области поиска станет больше её конца.

```

.....
NAIDEN:= FALSE;

```

```

NA:= 1;    // номер первого элемента области поиска
KO:= N;    // номер последнего элемента области поиска
REPEAT
    SR:=(KO-NA) DIV 2+NA;
    IF A[SR] = ISKOMOE
        THEN
            NAIDEN:=TRUE
        ELSE
            IF ISKOMOE > A[SR]
                THEN
                    NA:= SR+1
                ELSE
                    KO:= SR-1;
UNTIL NAIDEN OR (NA>KO);
IF NAIDEN
    THEN
        WRITELN('Элемент найден, его номер - ',SR)
    ELSE
        WRITELN('Элемент не найден');

```