



Сортировка массивов

Сортировка ***ShakerSort***

Var

l,j,x,n:integer;

a:array[0...50] of integer;

begin

for i:=1 **to** n **do** read(a[i]);

for i:=2 **to** n **do**

for j:=n **downto** i **do**

if a[j-1]>a[j] **then**

begin

x:=a[j-1];

A[j-1]:=a[j];

A[j]:=x;

End;

writeln('Результат:');

For i:=1 **to** n **do** write (a[i]);

End.

- Запоминать, были или не были перестановки в процессе некоторого прохода.

- Запоминать не только сам факт, что обмен имел место, но и положение (индекс) последнего обмена.

- Чередовать направление последовательных просмотров.

- **Левая граница = Номер первого элемента**

- **Правая граница = Номер последнего элемента**

- **Пока Левая граница < Правой границы делать**

- **Прямой проход "Пузырька" отлевой границы до Правой-1**

- **Правая граница = Правая граница - 1**

- **Обратный проход "Пузырька" от Правой границы долевой+1**

- **Левая граница = Левая граница + 1**

-

ИСХОДНЫЙ МАССИВ:

33 7 15 11 0 4 25 -1

Первый шаг работы алгоритма:

33 7 15 11 0 4 25 -1 => **7 33** 15 11 0 4 25 -1.

7 **33 15** 11 0 4 25 -1 => 7 **15 33** 11 0 4 25 -1.

7 15 **33 11** 0 4 25 -1 => 7 15 **11 33** 0 4 25 -1

7 15 11 **33 0** 4 25 -1 => 7 15 11 **0 33** 4 25 -1

7 15 11 0 **33 4** 25 -1 => 7 15 11 0 **4 33** 25 -1

7 15 11 0 4 **33 25** -1 => 7 15 11 0 4 **25 33** -1

7 15 11 0 4 25 **33 -1** => **7 15 11 0 4 25 -1 33**

**Затем меняем направление просмотра и
продолжаем процесс**

Затем меняем направление просмотра и продолжаем процесс:

7 15 11 0 4 25 -1 33 => 7 15 11 0 4 **-1 25 33**

7 15 11 0 **4 -1 25 33** => 7 15 11 0 **-1 4 25 33**

7 15 11 **0 -1 4 25 33** => 7 15 11 **-1 0 4 25 33**

7 15 **11 -1 0 4 25 33** => 7 15 **-1 11 0 4 25 33**

7 **15 -1 11 0 4 25 33** => 7 **-1 15 11 0 4 25 33**

7 -1 15 11 0 4 25 33 => **-1 7 15 11 0 4 25 33**

Второй шаг работы алгоритма

-1 0 7 11 4 15 25 33

Второй шаг работы алгоритма

-1 0 4 7 11 15 25 33

k:= 25; {Индекс последнего изменения}

s:= 1; {Первый элемент массива}

e:= 25; {Последний элемент массива}

while e > s **do**

begin

for i:= e **downto** s+1 **do** if A [i] < A[i-1] **then**

begin

tmp := A[i];

A[i] := A[i-1];

A[i-1] := tmp;

k := i; {запоминание индекса последней перестановки}

end;

s:=k;

for i:= s **to** e-1 **do** if A[i]>A[i+1] **then**

begin

tmp := A[i];

A[i] := A[i+1];

A[i+1] := tmp;

k := i;

end;

e:=k;

end;

