**ГЛАВА 2. СОРТИРОВКА ДАННЫХ**

**2.1. Методы сортировки**

Сортировка массива – упорядочение элементов массива по возрастанию или убыванию.

Алгоритм сортировки состоит из:

1. сравнения, определяющего упорядоченность пары элементов;
2. перестановки, меняющей местами пару элементов.

**ПРИМЕР 2.1.** Упорядочить по возрастанию элементы данного действительного вектора.

**1 этап:** Постановка задачи – корректна

пусть n- количество элементов (размерность массива)

a(i)- элементы вектора

**2 этап:** Математическая модель:

И.Д. n  N, a(i) R, i=1,2,…n

В.Д. a(j)  R, j=1,2,…n

СВЯЗЬ: a(j)  {a(i): a(1)<a(2)<…<a(n)}, i,j=1,2,…n.

**3 этап:** Техническое задание:

|  |  |  |
| --- | --- | --- |
| *№* | *Условия на И.Д.* | *Результат выполнения программы* |
| 1 | n  N и a(i)  R, i=1,2,…n | Получить упорядоченную последовательность a(j) |
| 2 | n  N\R и a(i)  R, i=1,2,…n | Ввод заново значения n |
| 3 | n  R или a(i)  R, i=1,2,…n | ДСТ |

**4 этап:** Программное обеспечение – язык программирования TP 7.0.

**5 этап**: Тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№*  *правила* | *И.Д.* | | *Результат выполнения программы* | |
| **n** | **a(i)** | **a(j)** | *Сообщение* |
| 1 | 4 | (4, -5, 6, 3) | (-5, 3, 4, 6) | - |
| 1 | 3 | 3 | - |
| 2 | -1 | - | - | Ввод заново значения n |
| 1.5 | - | - |
| 3 | \* |  |  | ДСТ |
| 2 | (2, \*) |  |
| 3 | (\*, \*, \*) |  |

**6 этап**: Алгоритмизация

***начало***

Vvod n

**mass a(n)**

sort a(n)

Vivod a(n)

***конец***

**7 этап**: Листинг программы: *самостоятельно*

* ***Метод пузырька (сортировка обменами)***

Принцип метода пузырька:

1. Просматриваем последовательность чисел от конца к началу:
2. Сравниваем (n-1) и n элементы. Если они не расположены в порядке возрастания (убывания), то меняем их местами: значением n элемента будет значение (n-1), а значением (n-1) элемента будет значение n.

Если они расположены в порядке возрастания (убывания), то оставляем их на месте.

1. Сравниваем (n-2) и (n-1) элементы...
2. …

N. Сравниваем 1 и 2 элементы…

В результате просмотра получим упорядоченную таблицу, т.е. каждый следующий просмотр будет приводить к постановке очередного минимального элемента в левый конец рассматриваемой части таблицы.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Элементы** | **Перестановки** | | | | | | | | | |
|  | 8 | 8 | 8 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 3 | 3 | 3 | 1 | 8 | 8 | 8 | 2 | 2 | 2 | 2 |
|  | 6 | 6 | 1 | 3 | 3 | 3 | 2 | 8 | 8 | 3 | 3 |
|  | 1 | 1 | 6 | 6 | 6 | 2 | 3 | 3 | 3 | 8 | 6 |
|  | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 8 |

Sort

i:=2

i ≤ n

j := n

j ≥ i

a(j-1)>a(j)

C:=a(j-1);

a(j-1):=a(j);

a(j):=c

j := j-1

i := i+1

конец

1

1

+

+

+

–

–

–

* ***Минимаксный метод (сортировка выбором)***

Принцип минимаксного метода:

1. Среди элементов массива находим минимальный (максимальный) и меняем его местами с 1 элементом.
2. Среди оставшихся ищем минимальный (максимальный) и меняем его местами со 2 элементом.
3. …

N-1. Среди оставшихся ищем минимальный (максимальный) и меняем его местами c n элементом.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Элементы** | **Перестановки** | | | |
| 1. | 8 | 1 | 1 | 1 | 1 |
| **2.** | 3 | 3 | 2 | 2 | 2 |
| **3.** | 6 | 6 | 6 | 3 | 3 |
| **4.** | 1 min | 8 | 8 | 8 | 6 |
| **5.** | 2 | 2 min | 3 min | 6 min | 8 |

Sort

i:=1

i ≤ n-1

min:=a(i);

p := i

j ≤ n

min>a(j)

min:=a(j);

p:= j;

j := j+1

C:=a(i);

a(i):=a(p);

a(p):=c

конец

1

1

+

+

+

–

–

–

j:=i+1

i := i+1

**Практические задания:**

1. **Упорядочить по убыванию элементы данного действительного вектора. Какое условие в указанных выше алгоритмах нужно изменить, чтобы получить алгоритм сортировки по убыванию.** (задание к Главе 2 Сортировка данных, написать код программы)
2. **Подсчитать количество букв 'А' и 'В' в данном тексте. Вычеркнуть все буквы 'А' из текста, если количество букв 'В' больше букв 'А';** (Модуль String)
3. **Даны действительные числа s,t. Получить g(15,-s) + g(t,s2+25) - g(21s-1,st-t3)**

**где g(a,b) =**

(выполнить с помощью Процедуры и функции)

1. **Даны действительные a, b, c. Получить**

(выполнить с помощью Процедуры и функции)