

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6

з дисципліни

«Алгоритми і структури даних»

Виконав:

студент групи ІМ-43

Олексійчук Станіслав Юрійович

номер у списку групи: 23

Перевірила:

Молчанова А. А.

Київ 2024

Постановка задачі

1. Задано двовимірний масив (матрицю) цілих чисел $A[m,n]$ або $A[n,n]$, де m та n – натуральні числа (константи), що визначають розміри двовимірного масиву. Виконати сортування цього масиву або заданої за варіантом його частини у заданому порядку заданим алгоритмом (методом).

Сортування повинно бути виконано безпосередньо у двовимірному масиві «на тому ж місці», тобто без перезаписування масиву та/або його будь-якої частини до інших одно- або двовимірних масивів, а також без використання спискових структур даних.

2. Розміри матриці m та n взяти самостійно у межах від 7 до 10.

3. При тестуванні програми необхідно підбирати такі вхідні набори початкових значень матриці, щоб можна було легко відстежити коректність виконання сортування і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Варіант № 23:

Задано двовимірний масив (матрицю) цілих чисел $A[m,n]$. Відсортувати окремо кожен стовпчик масиву алгоритмом №1 методу вставки (з лінійним пошуком зліва) за незменшенням.

Текст програми

Для виконання цієї лабораторної було використано програму для матриці (двовимірного масиву), а саме з розмірами 10x8 (10 рядків і 8 стовпців). Ось приклад програми:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 10
#define LENGTH 8

void print_array(int matrix[SIZE][LENGTH]){
    for (int row = 0; row < SIZE; row++) {
        for (int column = 0; column < LENGTH; column++) {
            printf("%d\t", matrix[row][column]);
        }
        printf("\n");
    }
    printf("\n-----\n\n");
}

int main()
{
    int row;
    int column;

    int index;
    int additional_index;
    int another_index;

    int element;
```

```

int matrix[SIZE][LENGTH] = {
    { -99,  95,  -10,  50,  -16,  32,  14,  -25 },
    { -56,  81,  -15,  50,  -21,  19,  18,  38 },
    { -23,  63,  -36,  50,   10,  -5,  30,   0 },
    {   0,  42,  -44,  50,   23,  14,  42, -11 },
    {  11,  29,  -58,  50,   23,  -7,  52,   5 },
    {  32,  10,  -78,  50,  -39,  40,  58, -11 },
    {  70,   5,  -83,  50,   35,  -2,  69,  15 },
    {  87,   4,  -90,  50,  -10,  25,  75,  -8 },
    {  97,   3,  -95,  50,   10,  25,  82,   1 },
    {  98,   3,  -98,  50,   -6,  25,  93,   2 }
};

print_array(matrix);

for (column = 0; column < LENGTH; column++) {
    for (index = 1; index < SIZE; index++) {
        element = matrix[index][column];
        additional_index = 0;
        while (matrix[additional_index][column] < element) {
            additional_index++;
        }
        for (another_index = index - 1; another_index >=
additional_index; another_index--) {
            matrix[another_index + 1][column] =
matrix[another_index][column];
        }
        matrix[additional_index][column] = element;
    }
}

```

```
print_array(matrix);
```

```
return 0;
```

```
}
```

Результати тестування програми

Для наглядності тестування кожен стовпець відсортовано по-своєму (нумерацію стовпців позначатиму індексами):

- 1) Відсортований (0, 6 стовпці);
- 2) Обернено відсортований (1, 2 стовпці);
- 3) Відсортований за незменшенням і незбільшенням – константа (3 стовпець);
- 4) Невідсортований (4, 5, 7 стовпці).

-99	95	-10	50	-16	32	14	-25
-56	81	-15	50	-21	19	18	38
-23	63	-36	50	10	-5	30	0
0	42	-44	50	23	14	42	-11
11	29	-58	50	23	-7	52	5
32	10	-78	50	-39	40	58	-11
70	5	-83	50	35	-2	69	15
87	4	-90	50	-10	25	75	-8
97	3	-95	50	10	25	82	1
98	3	-98	50	-6	25	93	2

Початкова матриця

-99	3	-98	50	-39	-7	14	-25
-56	3	-95	50	-21	-5	18	-11
-23	4	-90	50	-16	-2	30	-11
0	5	-83	50	-10	14	42	-8
11	10	-78	50	-6	19	52	0
32	29	-58	50	10	25	58	1
70	42	-44	50	10	25	69	2
87	63	-36	50	23	25	75	5
97	81	-15	50	23	32	82	15
98	95	-10	50	35	40	93	38

Відсортована матриця

-99	95	-10	50	-16	32	14	-25
-56	81	-15	50	-21	19	18	38
-23	63	-36	50	10	-5	30	0
0	42	-44	50	23	14	42	-11
11	29	-58	50	23	-7	52	5
32	10	-78	50	-39	40	58	-11
70	5	-83	50	35	-2	69	15
87	4	-90	50	-10	25	75	-8
97	3	-95	50	10	25	82	1
98	3	-98	50	-6	25	93	2

-99	3	-98	50	-39	-7	14	-25
-56	3	-95	50	-21	-5	18	-11
-23	4	-90	50	-16	-2	30	-11
0	5	-83	50	-10	14	42	-8
11	10	-78	50	-6	19	52	0
32	29	-58	50	10	25	58	1
70	42	-44	50	10	25	69	2
87	63	-36	50	23	25	75	5
97	81	-15	50	23	32	82	15
98	95	-10	50	35	40	93	38

Process returned 0 (0x0) execution time : 0.150 s
Press any key to continue.

Загальний вигляд результату

Висновок: я зрозумів, що таке сортування, суть різних алгоритмів сортування, наприклад, вставкою, вибором, обміном, швидкого сортування, бульбашкового тощо; зрозумів, що найбільше алгоритми сортування використовуються в базах даних; навчився використовувати алгоритм сортування на тому самому місці – сортування вставкою з лінійним пошуком зліва для сортування стовпців, використовуючи двовимірний масив; зрозумів переваги та недоліки кожного алгоритму, де вони можуть використовуватися та який з них бажано найчастіше використовувати для своїх цілей.