

**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-42

Лобань Михайло Юрійович

номер у списку групи: 21

Перевірила:

Молчанова А. А.

Київ 2024

Постановка задачі

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.

Завдання для варіанту

Варіант № 21

Задано двовимірний масив (матрицю) цілих чисел $A[m,n]$.
Відсортувати елементи першого рядка масиву, що стоять на парних позиціях, алгоритмом №2 методу обмінів («бульбашкове сортування») з використанням «прапорця» за незменшенням.

Текст програми

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{

    srand(time(0));

    int m, n; //m rows and n columns
    printf("enter amount of rows and columns: ");
    scanf("%d %d", &m, &n);

    int matrix[m][n];

    int generate_randomly = 0;

    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            if (i == 0 && j % 2 == 0) {
                if(generate_randomly) {
                    matrix[i][j] = rand() % 40 - 20;
                } else {
```

```

        printf("enter element matrix[%d][%d]: ", i, j);
        scanf("%d", &matrix[i][j]);
    }
} else {
    matrix[i][j] = 0;
}

}

}

```

```

printf("\nmatrix:\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        if (i == 0 && j % 2 == 0) {
            printf("\x1b[31m%5d\x1b[0m", matrix[i][j]);
        } else {
            printf("%5d", matrix[i][j]);
        }
    }
    printf("\n");
}

```

```

//sort

```

```

int flag = 1;
int temp;

```

```

while(flag) {
    flag = 0;
    for (int i = 0; i + 2 < n; i += 2) {

```

```

        if (matrix[0][i] > matrix[0][i + 2]) {
            temp = matrix[0][i];
            matrix[0][i] = matrix[0][i + 2];
            matrix[0][i + 2] = temp;
            flag = 1;
        }
    }
}

//result output
printf("\nsorted matrix:\n");
for (int i = 0; i < m; i++) {
    for (int j = 0; j < n; j++) {
        if (i == 0 && j % 2 == 0) {
            printf("\x1b[31m%5d\x1b[0m", matrix[i][j]);
        } else {
            printf("%5d", matrix[i][j]);
        }
    }
    printf("\n");
}

return 0;
}

```

Результати тестування

```
enter amount of rows and columns: 7 8
enter element matrix[0][0]: 5
enter element matrix[0][2]: 5
enter element matrix[0][4]: 5
enter element matrix[0][6]: 19
```

matrix:

5	0	5	0	5	0	19	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

sorted matrix:

5	0	5	0	5	0	19	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Process returned 0 (0x0) execution time : 13.369 s
Press any key to continue.

```
enter amount of rows and columns: 9 8
enter element matrix[0][0]: 14
enter element matrix[0][2]: 3
enter element matrix[0][4]: 0
enter element matrix[0][6]: 0
```

matrix:

14	0	3	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

sorted matrix:

0	0	0	0	3	0	14	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Process returned 0 (0x0) execution time : 9.563 s
Press any key to continue.

enter amount of rows and columns: 10 10

matrix:

-16	0	0	0	-1	0	3	0	9	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

sorted matrix:

-16	0	-1	0	0	0	3	0	9	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Process returned 0 (0x0) execution time : 5.867 s
Press any key to continue.

enter amount of rows and columns: 10 10

matrix:

19	0	-10	0	11	0	3	0	-12	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

sorted matrix:

-12	0	-10	0	3	0	11	0	19	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Process returned 0 (0x0) execution time : 1.490 s
Press any key to continue.

```

enter amount of rows and columns: 8 10

matrix:
-13  0  5  0 -17  0 -8  0 10  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0

sorted matrix:
-17  0 -13  0 -8  0  5  0 10  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0

Process returned 0 (0x0)   execution time : 1.393 s
Press any key to continue.
|

```

Висновки

В результаті виконання завдання було створено програму для розв'язання задачі сортування елементів масиву методом бульбашкового сортування з використанням прапорця. У ході тестування програма продемонструвала стабільну роботу, забезпечуючи правильне виконання сортування для різних вхідних даних.