

**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 №2.2

з дисципліни
«Алгоритми і структури даних»

Виконав:

Студент групи ІМ-33
Набока Тимур Максимович
номер у списку групи: 18

Перевірила:

Молчанова А. А.

Київ 2023

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 № 18

Задано двовимірний масив (матрицю) цілих чисел $A[m,n]$. Відсортувати окремо кожен стовпчик масиву методом швидкого сортування (методом Хоара) за незбільшенням.

Текст програми

```
#include <stdio.h>

int quicksort(int arr[], int low, int high) {
    if (low < high) {
        int pivotIndex = low;
        int pivot = arr[high];
        for (int i = low; i < high; i++) {
            if (arr[i] <= pivot) {

                int temp = arr[i];
                arr[i] = arr[pivotIndex];
                arr[pivotIndex] = temp;
                pivotIndex++;
            }
        }

        int temp = arr[high];
        arr[high] = arr[pivotIndex];
        arr[pivotIndex] = temp;

        quicksort(arr, low, pivotIndex - 1);
        quicksort(arr, pivotIndex + 1, high);
    }
}

int main() {
    int matrix[7][8] = {
        {77, 18, 11, 28, 77, 46, 25, 70},
        {94, 29, 48, 83, 62, 48, 84, 32},
        {54, 18, 99, 83, 60, 25, 57, 72},
        {56, 94, 67, 97, 39, 39, 37, 36},
        {29, 74, 72, 80, 97, 85, 70, 11},
        {65, 19, 19, 53, 82, 51, 70, 48},
        {52, 35, 32, 24, 59, 35, 35, 42}
    };
};
```

```
printf("Original matrix:\n");
for (int i = 0; i < 7; i++) {
    for (int j = 0; j < 8; j++) {
        printf("%d\t", matrix[i][j]);
    }
    printf("\n");
}

for (int i = 0; i < 7; i++) {
    quicksort(matrix[i], 0, 7);
}

printf("\nSorted matrix:\n");
for (int i = 0; i < 7; i++) {
    for (int j = 0; j < 8; j++) {
        printf("%d\t", matrix[i][j]);
    }
    printf("\n");
}

return 0;
}
```

Тести

Original matrix:

77	18	11	28	77	46	25	70
94	29	48	83	62	48	84	32
54	18	99	83	60	25	57	72
56	94	67	97	39	39	37	36
29	74	72	80	97	85	70	11
65	19	19	53	82	51	70	48
52	35	32	24	59	35	35	42

Sorted matrix:

11	18	25	28	46	70	77	77
29	32	48	48	62	83	84	94
18	25	54	57	60	72	83	99
36	37	39	39	56	67	94	97
11	29	70	72	74	80	85	97
19	19	48	51	53	65	70	82
24	32	35	35	35	42	52	59

PS C:\Users\user\Desktop\КПІ матеріали\Лабка на АСД\Лаба 6 (2.2)> █