

**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»

Виконала:

студентка групи ІМ-43
Козаченко Софія Олександрівна
номер у списку групи: 14

Перевірила:

Молчанова А.А.

Київ 2024

Завдання:

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 19 протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру

Варіант № 14

Задано квадратну двовимірний масив (матрицю) цілих чисел $A[n,n]$. Відсортувати головну діагональ масиву алгоритмом №4 методу обмінів («шейкерне сортування») за незбільшенням.

Текст програми :

```
#include <stdio.h>

#include <stdlib.h>

int main(void) {

    int A [8][8] = {

        {2,1,1,1,1,1,1,1},

        {1,7,1,1,1,1,1,1},
```

```
    {1,1,3,1,1,1,1,1},  
    {1,1,1,10,1,1,1,1},  
    {1,1,1,1,2,1,1,1},  
    {1,1,1,1,1,9,1,1},  
    {1,1,1,1,1,1,7,1},  
    {1,1,1,1,1,1,1,6},  
};  
  
printf("Matrix is:\n");  
for (int i = 0; i < 8; i++) {  
    for (int j = 0; j < 8; j++) {  
        printf("%4d", A[i][j]);  
    }  
    printf("\n");  
}  
  
printf("\n");
```

```
int F, i;  
int L = 0;  
int n =8;  
int R = n - 1;  
int T;  
  
do {  
    F = 0;
```

```
    for (i = L; i < R; i++) {

        if (A[i][i] < A[i+1][i+1]) {

            T = A[i][i];

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

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

            F = 1;

        }

    }

    R--;

    for (i = R; i > L; i--) {

        if (A[i][i] > A[i-1][i-1]) {

            T = A[i][i];

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

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

            F = 1;

        }

    }

    L++;

}

while (L < R && F == 1);

printf("New Matrix is:\n");

for (int i = 0; i < 8; i++) {
```

```

        for (int j = 0; j < 8; j++) {

            printf("%4d", A[i][j]);

        }

        printf("\n");

    }

    printf("\n");

    return 0;

}

```

Скріншоти тестів програми:

<p>Matrix is:</p> <pre> 10 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 8 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 4 </pre> <p>New Matrix is:</p> <pre> 10 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 8 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 4 </pre>	<p>Matrix is:</p> <pre> 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 8 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 9 </pre> <p>New Matrix is:</p> <pre> 9 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 8 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 </pre>	<p>Matrix is:</p> <pre> 2 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 10 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 6 </pre> <p>New Matrix is:</p> <pre> 10 1 1 1 1 1 1 1 1 9 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 7 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 </pre>
---	---	---

Висновок :

Виконавши лабораторну роботу, я засвоїла теоретичний матеріал з теми : «Алгоритми сортування» та набула практичних навичок рішення задачі сортування заданої категорії елементів за допомогою алгоритма

«Шейкерне сортування» у двовимірному масиві. Використання цього методу дозволило ефективно впорядковувати елементи головної діагоналі в матриці.

Зменшення кількості проходів, завдяки двонаправленому руху сортування по заданій матриці робить цей метод більш ефективним та швидким.

Лабораторна робота сприяла кращому розумінню принципів роботи алгоритмів сортування та їхньої адаптації до різних задач.