

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 №2.1
з дисципліни
«Алгоритми і структури даних»

Виконала:

студентка групи ІМ-21

Рабійчук Дар'я Олександрівна

номер у списку групи: 18

Перевірила:

Молчанова А.А.

Київ 2022

Постановка задачі :

- 1) Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) методом двійкового пошуку. Алгоритм двійкового пошуку задається варіантом завдання.
- 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 пошуку і ця коректність була б протестована для всіх можливих випадків. З метою тестування дозволяється використовувати матриці меншого розміру.

Варіант № 18

Задано матрицю дійсних чисел $A[m,n]$. Визначити присутність серед усіх елементів матриці будь-якого з чисел діапазону $[0,5]$ і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №2), якщо елементи кожного рядка окремо впорядковані за незбільшенням.

Текст програми :

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

int main()
{
    float matrix[7][8] = {
        {5, 4.9, 4.7, 3.2, 2.8, 2, 1, 1},
        {5, 5, 4.3, 4, 3, 2.9, 2.3, 1},
        {4.9, 4.3, 4, 4, 3.7, 3.3, 3, 1},
        {5, 4, 3, 2, 2, 2, 1, 1},
        {5, 5, 5, 4.7, 4, 3.8, 2.2, 1},
        {5, 4.7, 3.5, 2.6, 1.9, 1.5, 1.3, 1},
        {5, 4.4, 3.5, 3.1, 3, 2.1, 1.5, 1}
    };

    float searchNumber = 0;
    printf("Enter your number: ");
    scanf("%e", &searchNumber);

    int matrixLength = sizeof(matrix[0]) / sizeof(matrix[0][0]);
    int j = -1, i = 0;
    for(i; i < 7; i++){
        int L = 0, R = matrixLength - 1;
        while(L < R){
            int middleIndex = (L + R)/2;
            float middle = matrix[i][middleIndex];
            if(searchNumber < middle){
                L = middleIndex + 1;
            } else{
                R = middleIndex;
            }
        }
        if(R == L && matrix[i][R] == searchNumber && matrix[i][L] == searchNumber){
            j = L;
        }
        if (j != -1){
            break;
        }
    }
    if (j != -1){
        printf("row: %d; column: %d", i, j);
    } else {
        printf("no such element in matrix");
    }
    return 0;
}
```

Вхідні дані :

```
float matrix[7][8] = {
    {5, 4.9, 4.7, 3.2, 2.8, 2, 1, 1},
    {5, 5, 4.3, 4, 3, 2.9, 2.3, 1},
    {4.9, 4.3, 4, 4, 3.7, 3.3, 3, 1},
    {5, 4, 3, 2, 2, 2, 1, 1},
    {5, 5, 5, 4.7, 4, 3.8, 2.2, 1},
    {5, 4.7, 3.5, 2.6, 1.9, 1.5, 1.3, 1},
    {5, 4.4, 3.5, 3.1, 3, 2.1, 1.5, 1}
};
```

Результати тестування програми :

```
Enter your number: 4.3
row: 1; column: 2
Process returned 0 (0x0)   execution time : 4.366 s
Press any key to continue.
```

```
Enter your number: 1
row: 0; column: 6
Process returned 0 (0x0)   execution time : 1.201 s
Press any key to continue.
```

```
Enter your number: 5
row: 0; column: 0
Process returned 0 (0x0)   execution time : 1.356 s
Press any key to continue.
```

```
Enter your number: 3
row: 1; column: 4
Process returned 0 (0x0)   execution time : 1.554 s
Press any key to continue.
```

```
Enter your number: 3.5
row: 5; column: 2
Process returned 0 (0x0)   execution time : 2.787 s
Press any key to continue.
```

```
Enter your number: 2
row: 0; column: 5
Process returned 0 (0x0)   execution time : 0.908 s
Press any key to continue.
```

```
Enter your number: 6
no such element in matrix
Process returned 0 (0x0)   execution time : 6.851 s
Press any key to continue.
```