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
«Структури даних і алгоритми»

Виконав: Давидчук А.М.

студент групи ІО-41
Давидчук Артем Миколайович
номер у списку групи: 08

Перевірив:

Сергієнко А. М.

Київ 2024

Тема: Алгоритми двійкового пошуку.

Мета: Засвоєння теоретичного матеріалу та набуття практичних навичок розв'язання задачі пошуку заданої категорії елементів за допомогою різних алгоритмів методу двійкового пошуку у двовимірних масивах.

Завдання:
Написати програму розв'язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) методом двійкового пошуку. Алгоритм двійкового пошуку задається варіантом завдання.

Мій варіант:
Задано матрицю дійсних чисел A[m, n]. Окремо у кожному рядку матриці визначити присутність заданого дійсного числа X і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №2), якщо елементи кожного рядка окремо впорядковані за незменшенням

Код:

```
#include <stdio.h>

int main() {

    double x;
    int row, col;

    printf("Enter row, col numbers of matrix: ");
    scanf("%d, %d", &row, &col);

    double Matrix[row][col];

    printf("Enter matrix: \n");

    for (int i = 0; i < row; i++){
        for (int j = 0; j < col; j++){
            scanf("%lf", &Matrix[i][j]);
        }
    }

    printf("Enter x number: ");
    scanf("%lf", &x);

    for (int i = 0; i < row; i++) {

        int L = 0;
        int R = col-1;
```

```
        while (L < R) {

            int mid = (R + L) / 2;

            if (Matrix[i][mid] < x) L = mid+1;
            else R = mid;

        }

        if (Matrix[i][R] == x) printf("Element %.2lf is in row %d at
position %d\n", x, i, R);
        else printf("Element %.2lf is not in row %d\n", x, i);

    }

    return 0;

}
```

Тестування програми:

```
Enter row, col numbers of matrix: 7, 7
Enter matrix:
-5.3  -3.4  -1.2   0.0   2.1   3.7   5.0
-4.7  -2.5   0.0   0.0   0.0   3.3   5.5
-3.9  -1.1   0.0   2.1   3.5   5.7   7.3
-2.3   1.3   2.5   4.7   5.9   6.9   8.3
-1.5   0.0   1.3   3.4   5.7   7.9   8.4
 0.0   0.0   1.7   3.5   4.7   5.7   6.9
 0.0   2.1   3.3   4.5   6.7   8.9   9.0
Enter x number: 0.0
Element 0.00 is in row 0 at position 3
Element 0.00 is in row 1 at position 2
Element 0.00 is in row 2 at position 2
Element 0.00 is not in row 3
Element 0.00 is in row 4 at position 1
Element 0.00 is in row 5 at position 0
Element 0.00 is in row 6 at position 0

Enter row, col numbers of matrix: 8, 9
Enter matrix:
-4.5  -2.3  -1.2   0.0   1.5   3.1   4.7   5.3   6.7
-3.9  -1.7   0.0   0.0   0.0   2.1   3.5   5.0   6.3
-5.7  -3.5  -0.9   0.0   2.1   3.3   5.3   7.1   8.3
-3.5  -1.1   2.5   3.5   4.7   6.3   7.8   8.9   9.3
-2.9   0.0   0.0   1.3   3.5   4.9   5.7   6.9   7.3
-1.5   0.0   2.3   3.4   5.0   6.7   7.8   8.3   9.4
 0.0   1.1   2.3   3.3   4.5   5.5   7.0   8.3   9.1
 0.0   0.0   1.9   2.5   4.7   6.3   7.3   8.7   9.9
Enter x number: 0.0
Element 0.00 is in row 0 at position 3
Element 0.00 is in row 1 at position 2
Element 0.00 is in row 2 at position 3
Element 0.00 is not in row 3
Element 0.00 is in row 4 at position 1
Element 0.00 is in row 5 at position 1
Element 0.00 is in row 6 at position 0
Element 0.00 is in row 7 at position 0

Enter row, col numbers of matrix: 9, 10
Enter matrix:
-6.3  -4.7  -3.4  -2.1   0.0   2.3   3.5   5.0   6.7   8.3
-5.5  -3.5  -1.1   0.0   0.0   2.1   3.5   4.7   6.9   8.9
-3.3  -2.5  -0.9   0.0   1.7   3.4   5.7   7.3   8.4   9.5
-4.7  -3.1  -2.9  -1.3   1.1   2.7   4.5   6.3   7.9   8.5
-3.5  -2.1   0.0   0.0   2.1   3.9   5.7   6.9   8.3   9.0
-1.9   0.0   0.0   2.3   3.5   4.7   5.9   7.1   8.7   9.3
 0.0   1.5   2.7   3.4   4.5   5.3   7.0   8.1   8.9   9.9
-2.1  -1.5   0.0   2.3   3.5   5.0   6.7   7.8   8.5   9.7
 0.0   0.0   1.7   3.1   4.7   5.9   6.9   8.3   8.9   9.5
Enter x number: 0.0
Element 0.00 is in row 0 at position 4
Element 0.00 is in row 1 at position 3
Element 0.00 is in row 2 at position 3
Element 0.00 is not in row 3
Element 0.00 is in row 4 at position 2
Element 0.00 is in row 5 at position 1
Element 0.00 is in row 6 at position 0
Element 0.00 is in row 7 at position 2
Element 0.00 is in row 8 at position 0
```

Висновок:

Я реалізував алгоритм бінарного пошуку, який знаходить найлівіший релевантний елемент у вигляді методу обчислення, написаного на мові C. Продемонстрував успішність виконання під тестування програми.