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 №5

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. 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 14 :

Задано матрицю дійсних чисел A[m, n]. Окремо у першому рядку і останньому стовпчику визначити присутність заданого дійсного числа X і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №2), якщо елементи цього рядка і стовпчика впорядковані за незменшенням.

Текст програми :

```
#include <stdio.h>

int m = 8;

int n = 9;

   float A [8][9] = {

        {1,2,2,2,2,2,3,4,4},

        {1,1,1,1,1,1,1,1,4},

        {1,1,1,1,1,1,1,1,5},

        {1,1,1,1,1,1,1,1,5},
```

```
        {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},
        {1,1,1,1,1,1,1,1,7},
            };
printA () {
   printf("Matrix is:\n");
   for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 9; j++) {
            printf("%5.1f", A[i][j]);
        }
        printf("\n");
   }
   printf("\n");
}

int SearchX_Row(int x, int S) {
int L = 0;
int R = S - 1;
int b = -1;

int mid;
   while(L < R) {
```

```
        mid = (L + R) / 2;
        if(A[0][mid] == x) {
            R = mid;
            b = mid;
        } else if(A[0][mid] > x) {
            R = mid;
        } else if (A[0][mid] < x) {
            L = mid + 1;
        }
        if((L == R) && (A[0][S - 1])== x) {
             b = R;
        }
    }
        return b;
    }

int SearchX_Col(int x, int S) {
    int L = 0;
    int R = S - 1;
    int b = -1;

    int mid;
    while(L < R) {
```

```
        mid = (L + R) / 2;
        if(A[mid][S - 1] == x) {
            R = mid;
            b = mid;
        } else if(A[mid][S - 1] > x) {
            R = mid;
        } else if (A[mid][S - 1] < x) {
            L = mid + 1;
        }
        if((L == R) && (A[S - 1][8])== x) {
            b = R;
        }
    }
    return b;
}

int main(void) {
printA();
int res;
int size;

int X;
printf("Input X:");
```

```
scanf("%d", &X);


    res = SearchX_Row(X, m);
   if (res != -1) {
       printf("In first row, X: %d is found at (0,
%d)\n", X, res);
   } else {
       printf("In first row, X: %d is not found\n",
X);
   }


    res = SearchX_Col(X, n);
   if (res != -1) {
   printf("In last column, X: %d is found at (%d,
8)\n", X, res);
   } else {
       printf("In last column, X: %d is not found\n",
X);
   }


   return 0;
}
```

Скріншоти тестування :

```
Matrix is:
  1.0  2.0  2.0  2.0  2.0  2.0  3.0  4.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0

Input X:2
In first row, X: 2 is found at (0, 1)
In last column, X: 2 is not found
```

```
Matrix is:
  1.0  2.0  2.0  2.0  2.0  2.0  3.0  4.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0

Input X:4
In first row, X: 4 is found at (0, 7)
In last column, X: 4 is found at (0, 8)
```

```
Matrix is:
  1.0  2.0  2.0  2.0  2.0  2.0  3.0  4.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0

Input X:5
In first row, X: 5 is not found
In last column, X: 5 is found at (2, 8)
```

```
Matrix is:
  1.0  2.0  2.0  2.0  2.0  2.0  3.0  4.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0

Input X:10
In first row, X: 10 is not found
In last column, X: 10 is not found
```

```
Matrix is:
  1.0  2.0  2.0  2.0  2.0  2.0  3.0  4.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  4.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  5.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  6.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0
  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  7.0

Input X:6
In first row, X: 6 is not found
In last column, X: 6 is found at (3,8)
```

Висновок :

Виконавши лабораторну роботу, я засвоїла теоретичний матеріал з теми «Алгоритми двійкового пошуку» та набула практичних навичок реалізації алгоритму № 2 для пошуку заданого елемента у впорядкованому двовимірному масиві. Впорядкованість дозволила ефективно застосувати метод двійкового пошуку, що забезпечує значне скорочення кількості перевірок та прискорює знаходження елемента у порівнянні з лінійним пошуком.