Національний технічний університет України “Київський політехнічний інститут”

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп’ютерних систем

ЛАБОРАТОРНА РОБОТА №2.1

з дисципліни

“Структури даних і алгоритми”

ТЕМА: “АЛГОРИТМИ ДВІЙКОВОГО ПОШУКУ”

Група: КВ-34

Виконав : Фесенко Денис

Київ – 2024

Постановка задачі:

1. Написати програму розв’язання задачі пошуку (за варіантом) у двовимірному масиві (матриці) A[m][n] або A[n][n], в залежності від варіанту, одним з алгоритмів методу двійкового пошуку. Алгоритм двійкового пошуку задається варіантом завдання.

2. Пошук повинен бути виконаний 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. Розміри матриці m та n взяти самостійно у межах від 30 до 50.

Завдання за варіантом №23

Варіант № 23

Задано матрицю цілих чисел ***A[m][n]***. Окремо у останньому рядку і першому стовпчику визначити присутність будь-якого з чисел діапазону [**0**,**5**] і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №1), якщо елементи цього рядка і стовпчика впорядковані за незбільшенням.

Код програми

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define m 30
#define n 45

int main() {
  int A[m][n];

  int val = 1, i, j;
  for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
      A[m - 1 - i][n - 1 - j] = val;
      val++;
    }
  }
  for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
      printf("%d ", A[i][j]);
    }
    printf("\n");
  }

  for (int x = 0; x <= 5; x++) {
    int L = 0, R = m - 1;
    while (L <= R) { // col
      i = L + (R - L) / 2;

      if (x == A[i][0]) {
        break;
      } else {
        if (x > A[i][0])
          R = i - 1;
        else
          L = i + 1;
      }
    }
```

```
    if (L <= R) {
      printf("%d exists at the col pos: [%d][%d] ",
x, i, 0);
    } else {
      printf("%d doesnt exist at col ", x);
    }

    L = 0, R = n - 1;
    while (L <= R) { // row
      i = L + (R - L) / 2;

      if (x == A[m - 1][i]) {
        break;
      } else {
        if (x > A[m - 1][i])
          R = i - 1;
        else
          L = i + 1;
      }
    }
    if (L <= R) {
      printf("%d exists at the row pos: [%d][%d]",
x, m - 1, i);
    } else {
      printf("%d doesnt exist at row", x);
    }

    printf("\n");
  }
}
```

Тестування програми

1) m = n = 1

```
1
0 doesnt exist at col 0 doesnt exist at row
1 exists at the col pos: [0][0] 1 exists at the row pos: [0][0]
2 doesnt exist at col 2 doesnt exist at row
3 doesnt exist at col 3 doesnt exist at row
4 doesnt exist at col 4 doesnt exist at row
5 doesnt exist at col 5 doesnt exist at row
~
```

2) m = n = 3

```
9 8 7
6 5 4
3 2 1
0 doesnt exist at col 0 doesnt exist at row
1 doesnt exist at col 1 exists at the row pos: [2][2]
2 doesnt exist at col 2 exists at the row pos: [2][1]
3 exists at the col pos: [2][0] 3 exists at the row pos: [2][0]
4 doesnt exist at col 4 doesnt exist at row
5 doesnt exist at col 5 doesnt exist at row
```

3) m < n, а також m та n >30

```
0 doesnt exist at col 0 doesnt exist at row
1 doesnt exist at col 1 exists at the row pos: [34][44]
2 doesnt exist at col 2 exists at the row pos: [34][43]
3 doesnt exist at col 3 exists at the row pos: [34][42]
4 doesnt exist at col 4 exists at the row pos: [34][41]
5 doesnt exist at col 5 exists at the row pos: [34][40]
~
```

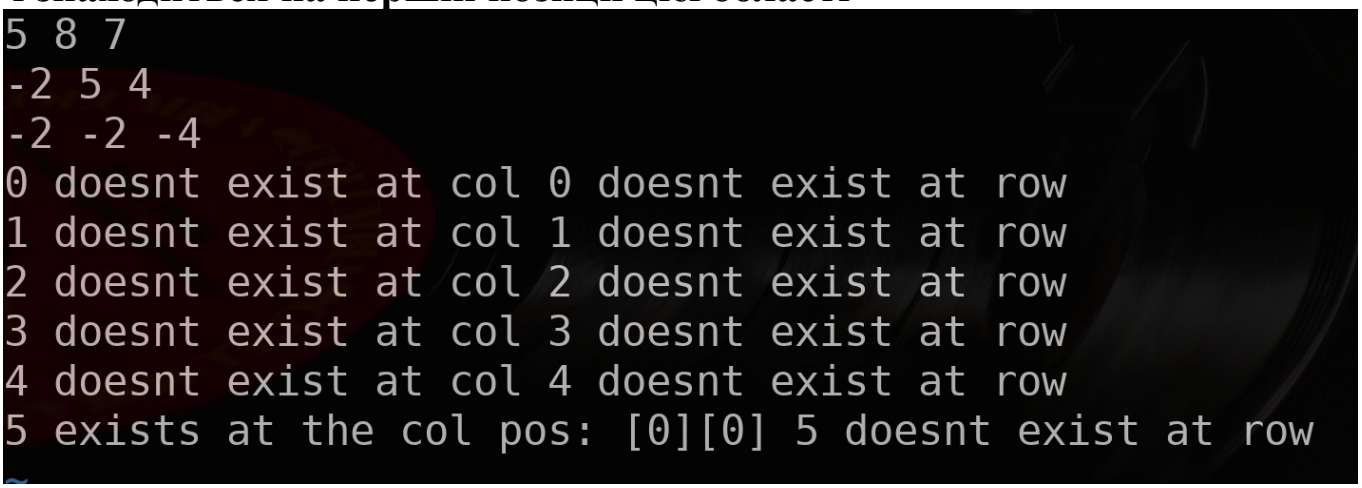
4) m > n, а також m та n >30

```
0 doesnt exist at col 0 doesnt exist at row
1 doesnt exist at col 1 exists at the row pos: [44][34]
2 doesnt exist at col 2 exists at the row pos: [44][33]
3 doesnt exist at col 3 exists at the row pos: [44][32]
4 doesnt exist at col 4 exists at the row pos: [44][31]
5 doesnt exist at col 5 exists at the row pos: [44][30]
```

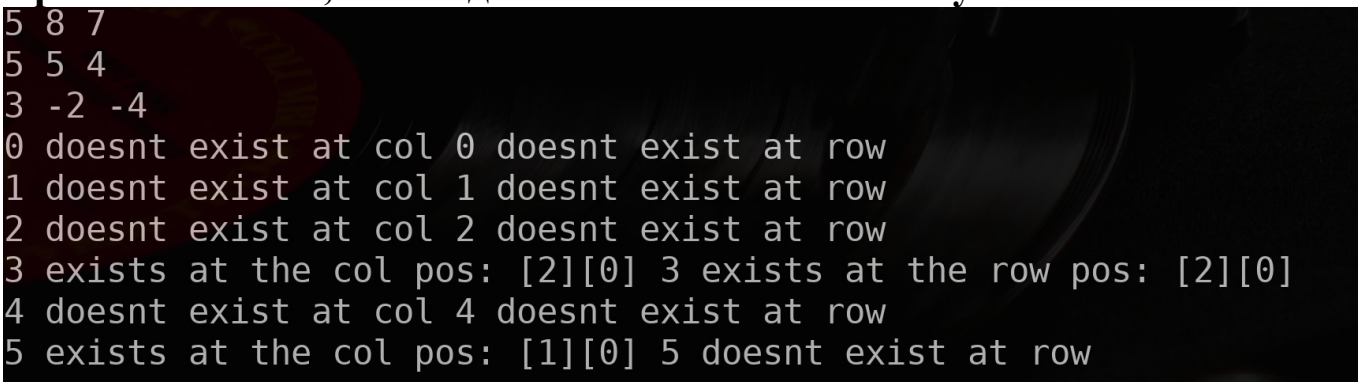
5) шуканий елемент відсутній у заданій за варіантом області

```
403 doesnt exist at col 403 doesnt exist at row
~
```

6)шуканий елемент присутній у заданій за варіантом області один раз і знаходиться на першій позиції цієї області



7)у лівій половині заданої за варіантом області знаходиться не менше трьох елементів, співпадаючих за значенням з шуканим



8)прямо по центру заданої за варіантом області знаходиться не менше трьох елементів, співпадаючих за значенням з шуканим