

**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 ПІ-05
Лавринович Марія Юріївна
номер у списку групи: 15

Перевірила:

Київ 2020

Постановка задачі

Написати програму розв'язання задачі пошуку (за варіантом) у двовірному масиві (матриці) методом двійкового пошуку.

Варіант № 15

Задано матрицю дійсних чисел $A[m,n]$. Окремо у останньому рядку і першому стовпчику визначити присутність заданого дійсного числа X і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №1), якщо елементи цього рядка і стовпчика впорядковані за незменшенням.

Текст програми

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

```
#define M 10
```

```
#define N 8
```

```
int matrix[M][N] = {  
    { -2, 1, 4, 7, 10, 13, 17, 20 },  
    { -2, 1, 4, 7, 10, 13, 17, 20 },  
    { -2, 1, 4, 7, 10, 14, 17, 20 },  
    { -1, 2, 5, 7, 11, 14, 17, 20 },  
    { -1, 2, 5, 8, 11, 14, 18, 21 },  
    { -1, 2, 5, 8, 11, 14, 18, 21 },  
    { -1, 2, 5, 8, 11, 15, 18, 21 },  
    { 0, 2, 5, 8, 11, 15, 18, 21 },  
    { 0, 2, 5, 8, 11, 15, 18, 21 },  
    { 0, 3, 6, 9, 12, 16, 19, 22 }};
```

```
int BinarSearch(int arr[M][N], int rows, int cols, int key)  
{  
    int first, last, mid, temp;  
    for (int i=0; i<rows; i++) {  
        first = 0;  
        last = cols - 1;  
        printf("last %d \n", last);
```

```

while (first < last) {

    mid = (first + last) / 2;
    printf("mid %d \n", mid);

    if (key <= arr[i][mid]) {
        last = mid;
        printf("Last: %d \n", last);
    }
    else {
        first = mid + 1;
        printf("First: %d \n", first);
    }
}
if ( arr[i][last] == key) {
    printf("Found value, coordinates: [%d, %d] \n", i,
last);
    break;
    goto stop;
} else {
    printf("value not found \n");
    goto stop;
}
stop: printf("Jumped to stop \n" );
}
}
int main(void) {

    BinarSearch(matrix, 10, 8, 16); //16-елемент, що задаємо для
пошуку//
    return 0;
}

```

Вхідні дані

```
int matrix[10][8] = {
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 14, 17, 20 },
    { -1, 2, 5, 7, 11, 14, 17, 20 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 3, 6, 9, 12, 16, 19, 22 }};
```

Результати тестування програми

Пошук елемента 16:

```
1 #include <stdio.h>
2
3 #define M 10
4 #define N 8
5
6 int matrix[10][8] = {
7     { -2, 1, 4, 7, 10, 13, 17, 20 },
8     { -2, 1, 4, 7, 10, 13, 17, 20 },
9     { -2, 1, 4, 7, 10, 14, 17, 20 },
10    { -1, 2, 5, 7, 11, 14, 17, 20 },
11    { -1, 2, 5, 8, 11, 14, 18, 21 },
12    { -1, 2, 5, 8, 11, 14, 18, 21 },
13    { -1, 2, 5, 8, 11, 15, 18, 21 },
14    { 0, 2, 5, 8, 11, 15, 18, 21 },
15    { 0, 2, 5, 8, 11, 15, 18, 21 },
16    { 0, 3, 6, 9, 12, 16, 19, 22 }};
17
18 int BinarSearch(int arr[M][N], int rows, int cols, int key)
19 {
20     int first, last, mid, temp;
21
22     for (int i=0; i<rows; i++) {
23         first = 0;
24         last = cols - 1;
25         printf("last %d \n", last);
26
27         while (first < last) {
28             mid = (first + last) / 2;
29             printf("mid %d \n", mid);
30
31             if (key <= arr[i][mid]) {
32                 last = mid;
33             }
34         }
35     }
36 }
```

```
last 7
mid 3
First: 4
mid 5
First: 6
mid 6
Last: 6
Value not found
Jumped to stop
last 7
mid 3
First: 4
mid 5
First: 6
mid 6
Last: 6
Value not found
Jumped to stop
last 7
mid 3
First: 4
mid 5
Last: 5
mid 4
First: 5
Found value, coordinates: [9, 5]
-----
Process exited after 0.06298 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Пошук елемента -1:

```

3 #define M 10
4 #define N 8
5
6 int matrix[10][8] = {
7     { -2, 1, 4, 7, 10, 13, 17, 20 },
8     { -2, 1, 4, 7, 10, 13, 17, 20 },
9     { -2, 1, 4, 7, 10, 14, 17, 20 },
10    { -1, 2, 5, 7, 11, 14, 17, 20 },
11    { -1, 2, 5, 8, 11, 14, 18, 21 },
12    { -1, 2, 5, 8, 11, 14, 18, 21 },
13    { -1, 2, 5, 8, 11, 15, 18, 21 },
14    { 0, 2, 5, 8, 11, 15, 18, 21 },
15    { 0, 2, 5, 8, 11, 15, 18, 21 },
16    { 0, 3, 6, 9, 12, 16, 19, 22 }};
17
18 int BinSearch(int arr[M][N], int rows, int
19 {
20     int first, last, mid, temp;
21
22     for (int i=0; i<rows; i++) {
23         first = 0;
24         last = cols - 1;
25         printf("last %d \n", last);
26
27         while (first < last) {
28             mid = (first + last) / 2;
29             printf("mid %d \n", mid);
30
31             if (key <= arr[i][mid]) {
32                 last = mid;
33                 printf("Last: %d \n", last);
34             }
35         }
36     }
37 }

```

```

last 7
mid 3
Last: 3
mid 1
Last: 1
mid 0
First: 1
Value not found
Jumped to stop
last 7
mid 3
Last: 3
mid 1
Last: 1
mid 0
Last: 0
Found value, coordinates: [3, 0]

-----
Process exited after 0.03412 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Пошук елемента 0:

```

#include <stdio.h>

#define M 10
#define N 8

int matrix[10][8] = {
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 14, 17, 20 },
    { -1, 2, 5, 7, 11, 14, 17, 20 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 3, 6, 9, 12, 16, 19, 22 }};

int BinSearch(int arr[M][N], int rows, int
{
    int first, last, mid, temp;

    for (int i=0; i<rows; i++) {
        first = 0;
        last = cols - 1;
        printf("last %d \n", last);

        while (first < last) {
            mid = (first + last) / 2;
            printf("mid %d \n", mid);

```

```

last 7
mid 3
Last: 3
mid 1
Last: 1
mid 0
First: 1
Value not found
Jumped to stop
last 7
mid 3
Last: 3
mid 1
Last: 1
mid 0
Last: 0
Found value, coordinates: [7, 0]

-----
Process exited after 0.061 seconds with return value 0
Для продолжения нажмите любую клавишу . . .

```

Пошук елемента 22:

```
#include <stdio.h>

#define M 10
#define N 8

int matrix[10][8] = {
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 13, 17, 20 },
    { -2, 1, 4, 7, 10, 14, 17, 20 },
    { -1, 2, 5, 7, 11, 14, 17, 20 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 14, 18, 21 },
    { -1, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 2, 5, 8, 11, 15, 18, 21 },
    { 0, 3, 6, 9, 12, 16, 19, 22 }
};

int BinarSearch(int arr[M][N], int rows, int cols, int key)
{
    int first, last, mid, temp;

    for (int i=0; i<rows; i++) {
        first = 0;
        last = cols - 1;
        printf("last %d \n", last);

        while (first < last) {
            mid = (first + last) / 2;
            printf("mid %d \n", mid);

            if (key <= arr[i][mid]) {
                last = mid;
            }
            else {
                first = mid + 1;
            }
        }

        if (key == arr[i][last]) {
            printf("Found value, coordinates: [%d, %d]\n", i, last);
            return 1;
        }
        else {
            printf("Value not found\n");
            return 0;
        }
    }

    printf("Jumped to stop\n");
    return 0;
}
```

last 7
mid 3
First: 4
mid 5
First: 6
mid 6
First: 7
Value not found
Jumped to stop
last 7
mid 3
First: 4
mid 5
First: 6
mid 6
First: 7
Value not found
Jumped to stop
last 7
mid 3
First: 4
mid 5
First: 6
mid 6
First: 7
Found value, coordinates: [9, 7]

Process exited after 0.05665 seconds with return value 0
Для продолжения нажмите любую клавишу . . .