

**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»

Виконав:

студент групи ІМ-43  
Черепов Олександр Павлович  
номер у списку групи: 32

Перевірила:

Молчанова А. А.

Київ 2024

## 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 № 2

Задано матрицю дійсних чисел  $A[m,n]$ . Окремо у кожному стовпчику матриці визначити присутність заданого дійсного числа  $X$  і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №2), якщо елементи кожного стовпчика окремо впорядковані за незбільшенням.

## Текст програми

```
#include <stdio.h>

int main(void) {
    int n, m, i, j, D, U, midIndex, foundValue, nullCounter = 0;
    double x;

    printf("Enter the size of your matrix (2 values separated by space): <rows collumns>\n");
    scanf("%d %d", &m, &n);

    if ((m<7 || m>10) || (n<7 || n>10)) {
        printf("Error! The size values should be between 7 and 10.");
        return 0;
    }

    double matrix[m][n];
    int values[n];
```

```

for (i=0; i<n; i++) {
    for (j=0; j<m; j++) {
        printf("Enter a value for element [%d][%d]: ", j, i);
        scanf("%lf", &matrix[j][i]);
        if (j>0 && matrix[j][i]>matrix[j-1][i]) {
            printf("Error! The values should be sorted in descending order.\n");
            printf("Element's [%d][%d] value was changed to %.2lf!\n", j, i, matrix[j-1][i]);
            matrix[j][i]=matrix[j-1][i];
        }
    }
}

printf("Input matrix:\n");
for (i=0; i<m; i++) {
    for (j=0; j<n; j++) {
        printf("%5.2lf\t", matrix[i][j]);
    }
    printf("\n");
}

printf("Enter the required value: ");
scanf("%lf", &x);

for (i=0; i<n; i++) {
    U = 0;
    D = m-1;
    foundValue = 0;
    while (U<D && foundValue==0) {
        if (matrix[U][i]==x) {
            foundValue = 1;
            values[i] = U;
        }
        else {
            midIndex = (U+D)/2;
            if (matrix[midIndex][i]>x) {
                U = midIndex+1;
            }
            else D = midIndex;
        }
    }
    if (foundValue==0) {
        if (matrix[D][i]==x) {
            values[i] = D;
        }
        else {
            values[i] = m+1;
            nullCounter++;
        }
    }
}

if (nullCounter!=n){
    printf("The elements were found in the following positions:\n");
    for (i=0; i<n; i++) {
        if (values[i]<m+1) printf("[%d][%d] ", values[i], i);
    }
}

```

```

    }
}
else printf("Error! There are no elements such as <%.2lf>.", x);

return 0;
}

```

## Тестування програми

### I тест (елемент всередині масиву):

```

D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: 33
The elements were found in the following positions:
[3][2]
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>_

```

### II тест (елемент на верхній межі масиву):

```

D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: 34
The elements were found in the following positions:
[0][3]
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>

```

### III тест (елемент на нижній межі масиву):

```

D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: 0
The elements were found in the following positions:
[6][3]
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>

```

#### IV тест (декілька елементів у різних стовпцях):

```
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: 15
The elements were found in the following positions:
[5][1] [3][3] [5][7]
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>
```

#### V тест (декілька елементів в одному стовпці):

```
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: 10
The elements were found in the following positions:
[3][4]
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>
```

#### VI тест (елемент не знайдено):

```
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>test1
Input matrix:
56.00  23.00  47.00  34.00  12.00  54.00  89.00  89.00  67.00
45.00  20.00  45.00  30.00  11.00  49.00  75.00  65.00  54.00
38.00  19.00  43.00  18.00  11.00  38.00  70.00  45.00  49.00
27.00  19.00  33.00  15.00  10.00  35.00  61.00  31.00  40.00
23.00  17.00  30.00  11.00  10.00  30.00  55.00  20.00  19.00
19.00  15.00  18.00  8.00  6.00  19.00  39.00  15.00  11.00
12.00  4.00  3.00  0.00  2.00  11.00  5.00  11.00  8.00
Enter the required value: -1
Error! There are no elements such as <-1.00>.
D:\education\EducationalWork\EducationalWork\ACД\000LABS\lab5\tests>_
```

## Висновок

На лабораторній роботі я розв'язував задачу по знаходженню елементу масиву, використовуючи алгоритм двійкового пошуку, за допомогою мови програмування C. Для успішного виконання завдання я використовував Алгоритм №2, який шукає найлівіший елемент (в моєму випадку – найвищий). Так як елементи стовпчиків відсортовані за незбільшенням за умовою, я порівнював значення середнього елементу із вхідним значенням та змінював область пошуку у відповідності до того, більший цей елемент чи менший, поки межі не зійшлися, або ж не було знайдено елемент зліва. Якщо елемент знайдено, то його індекс записується в інший масив, з якого потім циклом беруться значення та виводиться результат.

У ході лабораторної роботи розширив свої знання про алгоритми пошуку та, крім вже відомого мені алгоритму лінійного пошуку, познайомився із методом двійкового пошуку. Також навчився з ним працювати та використовувати його.

Отже, внаслідок цієї лабораторної роботи я дізнався про інший спосіб пошуку в масивах, який у деяких випадках може бути швидшим та більш оптимізованим, ніж лінійний пошук.