

**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 ІМ-41
Куц Анна Василівна номер
у списку групи: 11

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.

Завдання

Варіант № 11

Задано матрицю дійсних чисел $A[m,n]$. Визначити присутність серед усіх елементів матриці заданого дійсного числа X і його місцезнаходження (координати) методом двійкового пошуку (Алгоритм №1), якщо елементи кожного стовпчика окремо впорядковані за незменшенням.

Текст програми

```
#include <stdio.h>

int main()
{
    int m, n;
    printf("Enter the number of rows (m) and columns (n) (m,n):
\n");
    scanf("%d,%d", &m, &n);
    double matrix[m][n];

    printf("Enter elements of the matrix: \n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%lf", &matrix[i][j]);
        }
    }

    printf("Your matrix: \n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("%.3lf\t\t", matrix[i][j]);
        }
        printf("\n");
    }

    double x;
    printf("Enter x to find in your matrix: \n");
    scanf("%lf", &x);

    int found = 0;
    for (int j = 0; j < n; j++)
    {
        int L = 0,
            R = m - 1;
        while (L <= R)
```

```

{
    int mid = (L + R) / 2;
    if (matrix[mid][j] == x)
    {
        printf("Your x is located at (%d, %d)\n", mid, j);
        found = 1;
        break;
    }
    else
    {
        if (matrix[mid][j] > x)
        {
            R = mid - 1;
        }
        else if (matrix[mid][j] < x)
        {
            L = mid + 1;
        }
    }
}
if (found) break;
}
if (!found)
{
    printf("Your x is not located at the matrix\n");
}

return 0;
}

```

Тестування програми

```
"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"
Enter the number of rows (m) and columns (n) (m,n):
7,7
Enter elements of the matrix:
1.0 2.0 3.0 4.0 5.0 6.0 7.0
1.0 2.0 3.0 4.0 5.0 6.0 7.0
1.0 2.0 3.0 4.0 5.0 6.0 7.0
8.0 9.0 10.0 11.0 12.0 13.0 14.0
8.0 9.0 10.0 11.0 12.0 13.0 14.0
8.0 9.0 10.0 11.0 12.0 13.0 14.0
15.0 16.0 17.0 18.0 19.0 20.0 21.0
Your matrix:
1.000      2.000      3.000      4.000      5.000      6.000      7.000
1.000      2.000      3.000      4.000      5.000      6.000      7.000
1.000      2.000      3.000      4.000      5.000      6.000      7.000
8.000      9.000      10.000     11.000     12.000     13.000     14.000
8.000      9.000      10.000     11.000     12.000     13.000     14.000
8.000      9.000      10.000     11.000     12.000     13.000     14.000
15.000     16.000     17.000     18.000     19.000     20.000     21.000
Enter x to find in your matrix:
13
Your x is located at (3, 5)

Process finished with exit code 0
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"
Enter the number of rows (m) and columns (n) (m,n):
10,2
Enter elements of the matrix:
1 2
1 2
3 4
3 4
5 6
5 6
6 7
6 7
9 9
10.1 10.1
Your matrix:
1.000      2.000
1.000      2.000
3.000      4.000
3.000      4.000
5.000      6.000
5.000      6.000
6.000      7.000
6.000      7.000
9.000      9.000
10.100     10.100
Enter x to find in your matrix:
6
Your x is located at (7, 0)

Process finished with exit code 0
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"
```

```
Enter the number of rows (m) and columns (n) (m,n):
```

```
8,8
```

```
Enter elements of the matrix:
```

```
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5
```

```
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5
```

```
5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
```

```
5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
```

```
9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5
```

```
9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5
```

```
13.0 13.5 14.0 14.5 15.0 15.5 16.0 16.5
```

```
13.0 13.5 14.0 14.5 15.0 15.5 16.0 16.5
```

```
Your matrix:
```

1.000	1.500	2.000	2.500	3.000	3.500	4.000	4.500
1.000	1.500	2.000	2.500	3.000	3.500	4.000	4.500
5.000	5.500	6.000	6.500	7.000	7.500	8.000	8.500
5.000	5.500	6.000	6.500	7.000	7.500	8.000	8.500
9.000	9.500	10.000	10.500	11.000	11.500	12.000	12.500
9.000	9.500	10.000	10.500	11.000	11.500	12.000	12.500
13.000	13.500	14.000	14.500	15.000	15.500	16.000	16.500
13.000	13.500	14.000	14.500	15.000	15.500	16.000	16.500

```
Enter x to find in your matrix:
```

```
3
```

```
Your x is located at (1, 4)
```

```
Process finished with exit code 0
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"
```

```
Enter the number of rows (m) and columns (n) (m,n):
```

```
7,8
```

```
Enter elements of the matrix:
```

```
1.0 1.0 1.0 1.0 2.0 2.0 2.0 2.0
```

```
1.0 1.0 1.0 1.0 2.0 2.0 2.0 2.0
```

```
3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0
```

```
3.0 3.0 3.0 3.0 4.0 4.0 4.0 4.0
```

```
5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0
```

```
5.0 5.0 5.0 5.0 6.0 6.0 6.0 6.0
```

```
7.0 7.0 7.0 7.0 8.0 8.0 8.0 8.0
```

```
Your matrix:
```

1.000	1.000	1.000	1.000	2.000	2.000	2.000	2.000
1.000	1.000	1.000	1.000	2.000	2.000	2.000	2.000
3.000	3.000	3.000	3.000	4.000	4.000	4.000	4.000
3.000	3.000	3.000	3.000	4.000	4.000	4.000	4.000
5.000	5.000	5.000	5.000	6.000	6.000	6.000	6.000
5.000	5.000	5.000	5.000	6.000	6.000	6.000	6.000
7.000	7.000	7.000	7.000	8.000	8.000	8.000	8.000

```
Enter x to find in your matrix:
```

```
4
```

```
Your x is located at (3, 4)
```

```
Process finished with exit code 0
```

Enter the number of rows (m) and columns (n) (m,n):

10,9

Enter elements of the matrix:

1.111 1.112 1.113 1.114 1.115 1.116 1.117 1.118 1.119
2.211 2.212 2.213 2.214 2.215 2.216 2.217 2.218 2.219
3.311 3.312 3.313 3.314 3.315 3.316 3.317 3.318 3.319
4.411 4.412 4.413 4.414 4.415 4.416 4.417 4.418 4.419
5.511 5.512 5.513 5.514 5.515 5.516 5.517 5.518 5.519
6.611 6.612 6.613 6.614 6.615 6.616 6.617 6.618 6.619
7.711 7.712 7.713 7.714 7.715 7.716 7.717 7.718 7.719
8.811 8.812 8.813 8.814 8.815 8.816 8.817 8.818 8.819
9.911 9.912 9.913 9.914 9.915 9.916 9.917 9.918 9.919
10.111 10.112 10.113 10.114 10.115 10.116 10.117 10.118 10.119

Your matrix:

1.111	1.112	1.113	1.114	1.115	1.116	1.117	1.118	1.119
2.211	2.212	2.213	2.214	2.215	2.216	2.217	2.218	2.219
3.311	3.312	3.313	3.314	3.315	3.316	3.317	3.318	3.319
4.411	4.412	4.413	4.414	4.415	4.416	4.417	4.418	4.419
5.511	5.512	5.513	5.514	5.515	5.516	5.517	5.518	5.519
6.611	6.612	6.613	6.614	6.615	6.616	6.617	6.618	6.619
7.711	7.712	7.713	7.714	7.715	7.716	7.717	7.718	7.719
8.811	8.812	8.813	8.814	8.815	8.816	8.817	8.818	8.819
9.911	9.912	9.913	9.914	9.915	9.916	9.917	9.918	9.919
10.111	10.112	10.113	10.114	10.115	10.116	10.117	10.118	10.119

Enter x to find in your matrix:

10.119

Your x is located at (9, 8)

"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"

Enter the number of rows (m) and columns (n) (m,n):

7,8

Enter elements of the matrix:

1.123 1.234 1.345 1.456 2.567 2.678 2.789 2.890
1.123 1.234 1.345 1.456 2.567 2.678 2.789 2.890
3.001 3.112 3.223 3.334 4.445 4.556 4.667 4.778
3.001 3.112 3.223 3.334 4.445 4.556 4.667 4.778
5.123 5.234 5.345 5.456 6.567 6.678 6.789 6.890
5.123 5.234 5.345 5.456 6.567 6.678 6.789 6.890
7.001 7.112 7.223 7.334 8.445 8.556 8.667 8.778

Your matrix:

1.123	1.234	1.345	1.456	2.567	2.678	2.789	2.890
1.123	1.234	1.345	1.456	2.567	2.678	2.789	2.890
3.001	3.112	3.223	3.334	4.445	4.556	4.667	4.778
3.001	3.112	3.223	3.334	4.445	4.556	4.667	4.778
5.123	5.234	5.345	5.456	6.567	6.678	6.789	6.890
5.123	5.234	5.345	5.456	6.567	6.678	6.789	6.890
7.001	7.112	7.223	7.334	8.445	8.556	8.667	8.778

Enter x to find in your matrix:

568

Your x is not located at the matrix

Process finished with exit code 0

"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"

Enter the number of rows (m) and columns (n) (m,n):

8,8

Enter elements of the matrix:

1.111 1.222 2.333 2.444 3.555 3.666 4.777 4.888
1.111 1.222 2.333 2.444 3.555 3.666 4.777 4.888
5.999 6.111 6.222 6.333 7.444 7.555 8.666 8.777
5.999 6.111 6.222 6.333 7.444 7.555 8.666 8.777
9.888 9.999 10.111 10.222 11.333 11.444 12.555 12.666
9.888 9.999 10.111 10.222 11.333 11.444 12.555 12.666
13.777 13.888 14.999 14.111 15.222 15.333 16.444 16.555
13.777 13.888 14.999 14.111 15.222 15.333 16.444 16.555

Your matrix:

1.111	1.222	2.333	2.444	3.555	3.666	4.777	4.888
1.111	1.222	2.333	2.444	3.555	3.666	4.777	4.888
5.999	6.111	6.222	6.333	7.444	7.555	8.666	8.777
5.999	6.111	6.222	6.333	7.444	7.555	8.666	8.777
9.888	9.999	10.111	10.222	11.333	11.444	12.555	12.666
9.888	9.999	10.111	10.222	11.333	11.444	12.555	12.666
13.777	13.888	14.999	14.111	15.222	15.333	16.444	16.555
13.777	13.888	14.999	14.111	15.222	15.333	16.444	16.555

Enter x to find in your matrix:

1.111

Your x is located at (1, 0)

Process finished with exit code 0

"/Users/annkuts/University/АСД/Лабораторні/lab 2.1/lab 2.1"

Enter the number of rows (m) and columns (n) (m,n):

5,6

Enter elements of the matrix:

1 2 3 4 5 6
2 3 4 5 6 7
3 4 4 6 7 8
4 5 5 7 8 9
5 6 6 8 9 10

Your matrix:

1.000	2.000	3.000	4.000	5.000	6.000
2.000	3.000	4.000	5.000	6.000	7.000
3.000	4.000	4.000	6.000	7.000	8.000
4.000	5.000	5.000	7.000	8.000	9.000
5.000	6.000	6.000	8.000	9.000	10.000

Enter x to find in your matrix:

4

Your x is located at (3, 0)

Process finished with exit code 0

Висновок

Завдяки виконанню лабораторної роботи №2.1 я засвоїла теоретичний матеріал та набула практичних навичок рішення задачі пошуку заданої категорії елементів за допомогою різних алгоритмів методу двійкового пошуку у двовимірних масивах. Було створено програму за допомогою класичного бінарного пошуку по стовпцях (Алгоритм 1). Пошук саме по стовпцях є раціональним, оскільки в завданні мого варіанту є умова, що стовпці відсортовані за незменшенням. Для цього введено цикл `for` з лічильником `j`, що рухається від 0 до заданої користувачем кількості стовпців. Було ініціалізовано 2 змінні – `L`, що позначає лівий край, та `R`, що є правим краєм - кінцевим елементом рядку, який ми розглядаємо у масиві. В середині було створено цикл з передумовою `while`, який ітерує допоки `L <= R`. У внутрішньому циклі ініціалізовано середній індекс рядка елементу `mid = (L+R)/2`. Було введено умову виходу з циклу, якщо `[mid][j]` є шуканим `x`, інакше - 2 умови: якщо середній елемент `[mid][j]` більший за шуканий `x`, ми змінюємо діапазон від `L` до `mid - 1` (рухаємось вліво), якщо середній елемент `[mid][j]` менший за шуканий `x`, ми змінюємо діапазон від `mid+1` до `R` (рухаємось вправо). В умову був введений індикатор `found`. Якщо він дорівнює 0, то програма виводить на екран повідомлення, що шуканий `x` не розташований у матриці. Якщо `found = 1`, то відбувається вихід з циклу з лічильником, що зменшує кількість ітерацій. Шуканий алгоритм має складність $O(\log 2 m)$.

У деяких тестуваннях, до прикладу, картинці номер 3, видно, що програма шукає не найлівіший елемент, а той, який перший знайде при виконанні алгоритму. Середнім елементом стовпця є елемент 7([3][4]), це більше, ніж 3, тому зміщуємо наш пошук вліво від 0 до 2. Середнім елементом в цьому діапазоні є 1, тож перевірка умови, чи більше елемент, чи менше, не працює, і виконується умова, що елемент `[mid][j] == x`, саме тому програма виводить координати `[1][4]` замість `[0][4]`.