

Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации

государственное бюджетное образовательное учреждение

высшего профессионального образования

ордена Трудового Красного Знамени

“Московский технический университет связи и информатики”

Задача №3 по дисциплине

“СИАОД”

Выполнил студент

Группы БВТ1902

Ахмедов Х.М.

Москва 2021

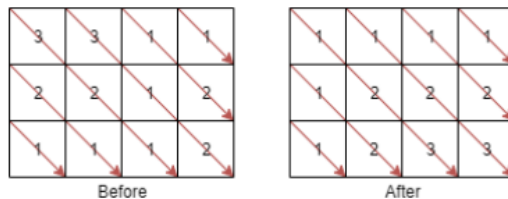
Оглавление

1. Задание на лабораторную работу	3
2. Листинг программы	3
3. Вывод.....	5

1. Задание на лабораторную работу

Задача 3. «Сортировка диагоналей в матрице»

Дана матрица `mat` размером $m * n$, значения - целочисленные. Напишите функцию, сортирующую каждую диагональ матрицы по возрастанию и возвращающую получившуюся матрицу.



Пример 3.1:

Ввод: `mat = [[3, 3, 1, 1], [2, 2, 1, 2], [1, 1, 1, 2]]`

Вывод: `[[1, 1, 1, 1], [1, 2, 2, 2], [1, 2, 3, 3]]`

Пример 3.2:

Ввод: `mat = [[11, 25, 66, 1, 69, 7], [23, 55, 17, 45, 15, 52], [75, 31, 36, 44, 58, 8], [22, 27, 33, 25, 68, 4], [84, 28, 14, 11, 5, 50]]`

Вывод: `[[5, 17, 4, 1, 52, 7], [11, 11, 25, 45, 8, 69], [14, 23, 25, 44, 58, 15], [22, 27, 31, 36, 50, 66], [84, 28, 75, 33, 55, 68]]`

Ограничения:

- $m == \text{len}(\text{mat})$
- $n == \text{len}(\text{mat}[i])$
- $1 \leq m, n \leq 100$
- $1 \leq \text{mat}[i][j] \leq 100$

2. Листинг программы

```
package Koursovaya;

import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

public class matrix {

    static void sortMatrix(int[][] mat, int m, int n) {

        for(int i = 0; i < m + n - 1; ++i) {
            int startX, startY;
            if(i < n){
                startX = 0;
                startY = n - i - 1;
            }else {
                startX = i - n;
                startY = 0;
            }
        }
    }
}
```

```

    }

    ArrayList<Integer> list = new ArrayList();

    int offset;
    int x;
    int y;
    for(offset = 0; offset <= 0; ++offset) {
        x = startX + offset;
        y = startY + offset;
        if (x >= m || y >= n) {
            break;
        }

        list.add(mat[x][y]);
    }

    list.sort(Integer::compareTo);

    for(offset = 0; offset <= 0; ++offset) {
        x = startX + offset;
        y = startY + offset;
        if (x >= m || y >= n) {
            break;
        }

        mat[x][y] = list.remove(0);
    }
}

static void printMatrix(int[][] mat, int m, int n) {
    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            System.out.print(mat[i][j] + " ");
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    Scanner in = new Scanner(System.in);

    System.out.println("Введите количество строк: ");
    int m = in.nextInt();
    System.out.println("Введите количество столбцов: ");
    int n = in.nextInt();

    int[][] mat = new int[m][n];
    Random rng = new Random();

    for(int i = 0; i < m; ++i) {
        for(int j = 0; j < n; ++j) {
            mat[i][j] = rng.nextInt(m*n + 1);
        }
    }

    System.out.println("Исходная матрица:");
    printMatrix(mat, m, n);
    sortMatrix(mat, m, n);
    System.out.println("Отсортированная матрица:");
    printMatrix(mat, m, n);
}
}

```

3. Вывод

Мы написали программу, которая сортирует каждую диагональ матрицы по возрастанию.