

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 №8 по дисциплине

“ Структуры и алгоритмы обработки данных”

Выполнил студент

Группы БФИ1901

Гасанов Г. М.

Москва 2021

## **Оглавление**

1. Задание на лабораторную работу .....	3
2. Листинг программы .....	3
3. Вывод .....	4

## 1. Задание на лабораторную работу

### Задача 1. «Шарики и стрелы»

Некоторые сферические шарики распределены по двумерному пространству. Для каждого шарика даны  $x$ -координаты начала и конца его горизонтального диаметра. Так как пространство двумерно, то  $y$ -координаты не имеют значения в данной задаче. Координата  $x_{start}$  всегда меньше  $x_{end}$ .

Стрелу можно выстрелить строго вертикально (вдоль  $y$ -оси) из разных точек  $x$ -оси. Шарик с координатами  $x_{start}$  и  $x_{end}$  уничтожается стрелой, если она была выпущена из такой позиции  $x$ , что  $x_{start} \leq x \leq x_{end}$ . Когда стрела выпущена, она летит в пространстве бесконечное время (уничтожая все шарики на пути).

Дан массив `points`, где `points[i] = [xstart, xend]`. Напишите функцию, возвращающую минимальное количество стрел, которые нужно выпустить, чтобы уничтожить все шарики.

#### Пример 1.1:

Ввод: `points = [[10,16],[2,8],[1,6],[7,12]]`

Вывод: 2

#### Пример 1.2:

Ввод: `points = [[1,2],[3,4],[5,6],[7,8]]`

Вывод: 4

#### Пример 1.3:

Ввод: `points = [[1,2],[2,3],[3,4],[4,5]]`

Вывод: 2

#### Пример 1.4:

Ввод: `points = [[1,2]]`

Вывод: 1

#### Пример 1.5:

Ввод: `points = [[2,3],[2,3]]`

Вывод: 1

#### Ограничения:

- $0 \leq \text{len}(\text{points}) \leq 10^4$
- $\text{len}(\text{points}[i]) == 2$
- $-2^{31} \leq x_{start} < x_{end} \leq 2^{31} - 1$

## 2. Листинг программы

```
package com.company;
import java.util.*;

public class Zad4 {

    public static int arrowsFind(int[][] points) {
        if (points.length == 0)
            return 0;
    }
}
```

```

Arrays.sort(points, (a, b) -> Integer.compare(a[1], b[1]));
int arrowCount = 0;
long end = Long.MIN_VALUE;
for (int [] p: points) {
    if (p[0] > end) {
        end = p[1];
        arrowCount += 1;
    }
}
return arrowCount;
}

public static void main (String[]args) {
    int [][] points1 = {{10,16},{2,8},{1,6},{7,12}};
    System.out.println(arrowsFind(points1));
    int [][] points2 = {{1,2},{3,4},{5,6},{7,8}};
    System.out.println(arrowsFind(points2));
    int [][] points3 = {{1,2},{2,3},{3,4},{4,5}};
    System.out.println(arrowsFind(points3));
    int [][] points4 = {{1,2}};
    System.out.println(arrowsFind(points4));
    int [][] points5 = {{2,3},{2,3}};
    System.out.println(arrowsFind(points5));
}
}

```

### 3. Вывод

Мы написали программу, которая возвращает минимальное количество стрел, которое нужно выпустить, чтобы уничтожить все шарики.