

60、演唱会，计算最多能观看多少场演出——区间问题/最多不想交区间个数

题目描述

为了庆祝中国共产党成立100周年，某公园将举行多场文艺表演，很多演出都是同时进行，一个人只能同时观看一场演出，且不能迟到早退，由于演出分布在不同的演出场地，所以连续观看的演出最少有15分钟的时间间隔，

小明是一个狂热的文艺迷，想观看尽可能多的演出，现给出演出时间表，请帮小明计算他最多能观看几场演出。

输入描述

第一行为一个数N，表示演出场数， $1 \leq N \leq 1000$ ，接下来N行，每行有被空格分割的两个整数，

第一个整数T表示演出的开始时间，第二个整数L表示演出的持续时间，T和L的单位为分钟， $0 \leq T \leq 1440, 0 < L \leq 100$ 。

输出描述

输出最多能观看的演出场数。

用例

输入	2 720 120 840 120
输出	1
说明	两场演出间隔时间为0，不满足最小15分钟时间间隔的要求，所以最多只能观看一场演出

输入	2 0 60 90 60
输出	2
说明	两场演出间隔大于15分钟，都能观看到

题目解析

本题其实是区间问题。且是要求解**最多不相交区间个数**。

需要注意的是，两个区间不相交的条件也比较特殊：

连续观看的演出最少有15分钟的时间间隔

即两个区间之间需要间隔15单位长度才算不相交。

JavaScript算法源码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  const lines = [];
10 let n;
11 rl.on("line", (line) => {
12   lines.push(line);
13
14   if (lines.length === 1) {
15     n = lines[0].trim();
16   }
17
18   if (n && lines.length === n + 1) {
19     lines.shift();
20     const matrix = lines.map((line) => line.split(" ").map(Number));
21     console.log(getResult(matrix));
22     lines.length = 0;
23   }
24 });
25
26 function getResult(matrix) {
27   matrix = matrix
```

```
27   matrix = matrix
28     .map(([start, spend]) => [start, start + spend])
29     .sort((a, b) => a[1] - b[1]);
30
31   let t = matrix[0][1];
32   let ans = 1;
33   for (let i = 1; i < matrix.length; i++) {
34     const [l, r] = matrix[i];
35
36     if (l - t >= 15) {
37       ans++;
38       t = r;
39     }
40   }
41
42   return ans;
43 }
```

Java算法源码

```
1  import java.util.Arrays;
2  import java.util.Scanner;
3
4  public class Main {
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7
8          int n = sc.nextInt();
9
10         int[][] ranges = new int[n][2];
11         for (int i = 0; i < n; i++) {
12             ranges[i][0] = sc.nextInt(); // 开始时间
13             ranges[i][1] = ranges[i][0] + sc.nextInt(); // 结束时间 = 开始时间 + 持续时间
14         }
15
16         System.out.println(getResult(ranges));
17     }
18
19     public static int getResult(int[][] ranges) {
20         Arrays.sort(ranges, (a, b) -> a[1] - b[1]);
21
22         int t = ranges[0][1];
23         int ans = 1;
24
25         for (int i = 1; i < ranges.length; i++) {
26             int l = ranges[i][0];
27             int r = ranges[i][1];
28
29             if (l - t >= 15) {
30                 ans++;
31                 t = r;
32             }
33         }
34
35         return ans;
36     }
37 }
```

Python算法源码

```
1 # 输入获取
2 n = int(input())
3 tmp = [list(map(int, input().split())) for _ in range(n)]
4 rans = list(map(lambda ran: [ran[0], ran[0] + ran[1]], tmp))
5
6
7 # 算法入口
8 def getResult():
9     rans.sort(key=lambda ran: ran[1])
10
11     t = rans[0][1]
12     ans = 1
13
14     for i in range(1, len(rans)):
15         l, r = rans[i]
16         if l - t >= 15:
17             ans += 1
18             t = r
19
20     return ans
21
22
23 # 算法调用
24 print(getResult())
```