

【华为OD机考 统一考试机试C卷】绘图机器 (C++ Java JavaScript Python C语言)

华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷

目前在考C卷，经过两个月的收集整理，**C卷真题已基本整理完毕**

抽到原题的概率为2/3到3/3，**也就是最少抽到两道原题。请注意：大家刷完C卷真题，最好要把B卷的真题刷一下，因为C卷的部分真题来自B卷。**

另外订阅专栏还可以联系笔者开通在线 **OJ** 进行刷题，提高刷题效率。

真题目录：华为OD机考机试 真题目录 (C卷 + D卷 + B卷 + A卷) + 考点说明

专栏：2023华为OD机试(B卷+C卷+D卷) (C++JavaJSPy)

华为OD面试真题精选：华为OD面试真题精选

在线OJ：点击立即刷题，模拟真实机考环境

题目描述

绘图机器的绘图笔初始位置在坐标(0,0)机器启动后按照以下规则来进行绘制直线。

1. 尝试沿着横轴正向绘制直线直到给定的终点E
2. 期间可以通过指令在纵坐标轴方向进行偏移，offsetY为正数表示正向偏移,为负数表示负向偏移

给定的横坐标终点值E 以及若干条绘制指令，

请计算绘制的直线和横坐标轴以及x=E的直线组成的图形面积。

输入描述

- 首行为两个整数 N 和 E
- 表示有N条指令,机器运行的横坐标终点值E
- 接下来N行 每行两个整数表示一条绘制指令 **x offsetY**
- 用例保证横坐标x以递增排序的方式出现

- 且不会出现相同横坐标x

取值范围

- $0 < N \leq 10000$
- $0 \leq x \leq E \leq 20000$
- $-10000 \leq \text{offsetY} \leq 10000$

输出描述

- 一个整数表示计算得到的面积 用例保证结果范围在0到4294967295之内。

用例1

输入

1	4	10
2	1	1
3	2	1
4	3	1
5	4	-2

输出

1	12
---	----

用例2

输入

1	2	4
2	0	1
3	2	-2

输出

C++

```
1  #include <iostream>
2  #include <vector>
3  #include <cmath>
4  using namespace std;
5
6  int main() {
7      int n, e;
8      cin >> n >> e;
9
10     if (e == 0) {
11         cout << 0 << endl;
12         return 0;
13     }
14
15     vector<int> offsets(e);
16
17     for (int i = 0; i < n; i++) {
18         int cur_x, offset_y;
19         cin >> cur_x >> offset_y;
20         offsets[cur_x] = offset_y;
21     }
22
23     vector<int> dp(e);
24     dp[0] = offsets[0];
25     for (int i = 1; i < e; i++) {
26         dp[i] = offsets[i] + dp[i - 1];
27     }
28
29     int ans = 0;
30     for (int num : dp) {
31         ans += abs(num);
32     }
33     cout << ans << endl;
34
35
36
```

```
    return 0;
}
```

java

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         int n = scanner.nextInt(); // 输入指令的数量
8         int e = scanner.nextInt(); // 终点横坐标
9
10        if (e == 0) { // 如果终点横坐标为0
11            System.out.println(0); // 输出面积为0
12            return;
13        }
14
15        int[] offsets = new int[e]; // 创建一个长度为终点横坐标的整数数组，用于存储纵坐标偏移量
16
17        for (int i = 0; i < n; i++) {
18            int cur_x = scanner.nextInt(); // 当前点的横坐标
19            int offset_y = scanner.nextInt(); // 当前点纵坐标相较于上一个点纵坐标的偏移量
20            offsets[cur_x] = offset_y; // 将偏移量存储在对应横坐标位置上
21        }
22
23        int[] dp = new int[e]; // 创建一个长度为终点横坐标的整数数组，用于存储每个横坐标位置的纵坐标偏移量之和
24        dp[0] = offsets[0]; // 第一个位置的纵坐标偏移量为指令中的纵坐标偏移量
25        for (int i = 1; i < e; i++) { // 从第二个位置开始遍历
26            dp[i] = offsets[i] + dp[i - 1]; // 当前位置的纵坐标偏移量为指令中的纵坐标偏移量加上前一个位置的纵坐标偏移量之和
27        }
28
29        int ans = 0; // 初始化面积为0
30        for (int num : dp) { // 遍历每个横坐标位置的纵坐标偏移量之和
31            ans += Math.abs(num); // 将绝对值加到面积中
32        }
33        System.out.println(ans); // 输出面积
34
35        scanner.close(); // 关闭输入流
36    }
37 }
```

```
36 |     }
37 | }
```

JavaScript

```
1  const readline = require('readline');
2  const rl = readline.createInterface({
3    input: process.stdin,
4    output: process.stdout
5  });
6
7  let lines = [];
8  let n = 0, e = 0;
9
10 rl.on('line', (line) => {
11   lines.push(line);
12
13   if (lines.length === 1) {
14     [n, e] = line.split(' ').map(Number);
15     if (e === 0) {
16       console.log(0);
17       lines = [];
18     }
19   }
20   if (e !== 0 && n !== 0 && lines.length === n + 1) {
21     lines.shift();
22
23     // 求出每个横轴单位上的offsetY偏移值, 如果输入未给定offsetY, 则相当于offsetY=0
24     const offsets = new Array(e).fill(0);
25     for (const line of lines) {
26       const [x, offsetY] = line.split(' ').map(Number);
27       offsets[x] = offsetY;
28     }
29
30     const dp = new Array(e).fill(0);
31     dp[0] = offsets[0];
32     for (let i = 1; i < e; i++) {
33       dp[i] = offsets[i] + dp[i - 1];
34     }
35
36     ~
```

```

36     let ans = 0;
37     for (const num of dp) {
38         ans += Math.abs(num);
39     }
40     console.log(ans);
41
42     lines = [];
43 }
44 });

```

python

```

1  n , e = map(int, input().split())
2
3
4  if e == 0: # 如果终点横坐标为0
5      print(0) # 输出面积为0
6      exit()
7
8  offsets = [0] * e # 创建一个长度为终点横坐标的整数数组, 用于存储纵坐标偏移量
9
10 for _ in range(n):
11     cur_x , offset_y = map(int, input().split())
12     offsets[cur_x] = offset_y # 将偏移量存储在对应横坐标位置上
13
14 dp = [0] * e # 创建一个长度为终点横坐标的整数数组, 用于存储每个横坐标位置的纵坐标偏移量之和
15 dp[0] = offsets[0] # 第一个位置的纵坐标偏移量为指令中的纵坐标偏移量
16 for i in range(1, e): # 从第二个位置开始遍历
17     dp[i] = offsets[i] + dp[i - 1] # 当前位置的纵坐标偏移量为指令中的纵坐标偏移量加上前一个位置的纵坐标偏移量之和
18
19 ans = 0 # 初始化面积为0
20 for num in dp: # 遍历每个横坐标位置的纵坐标偏移量之和
21     ans += abs(num) # 将绝对值加到面积中
22 print(ans) # 输出面积

```

C语言

```

1  #include <stdio.h>
2  #include <stdlib.h>
3

```

```

4
5 // 定义绘图指令的结构体
6 typedef struct {
7     int x;          // 横坐标
8     int offsetY;    // 纵坐标偏移
9 } Command;
10
11 int main() {
12     int N, E;
13     scanf("%d %d", &N, &E); // 读取指令数量和终点横坐标E
14
15     Command commands[N + 1]; // 存储绘图指令, 多一个位置用于终点
16
17     for (int i = 0; i < N; i++) {
18         scanf("%d %d", &commands[i].x, &commands[i].offsetY); // 读取每条指令
19     }
20
21     // 添加终点作为最后一条指令
22     commands[N].x = E;
23     commands[N].offsetY = 0;
24
25     unsigned long area = 0; // 总面积
26     int currentY = 0;      // 当前纵坐标偏移
27
28     for (int i = 0; i < N; i++) {
29         int base = commands[i + 1].x - commands[i].x; // 底边长度
30         int height = currentY + commands[i].offsetY; // 高度
31         area += base * abs(height); // 计算梯形面积
32
33         currentY += commands[i].offsetY; // 更新当前纵坐标偏移
34     }
35
36     printf("%lu\n", area); // 输出总面积
37     return 0;
38 }

```

完整用例

用例1

1	4	10
2	1	1
3	2	1
4	3	1
5	4	-2

用例2

1	2	4
2	0	1
3	2	-2

用例3

1	5	10
2	1	1
3	2	2
4	3	3
5	4	4
6	5	-5

用例4

1	3	6
2	0	1
3	2	-2
4	4	4

用例5

1	2	8
2	1	1
3	3	-3

用例6

1	4	10
2	0	2
3		

4	2	-2
5	4	4
6	6	-6

用例7

1	10	20
2	1	1
3	2	2
4	3	3
5	4	4
6	5	5
7	6	6
8	7	7
9	8	8
10	9	9
11	10	-10

用例8

1	6	15
2	0	1
3	2	-2
4	4	4
5	6	-6
6	8	8
7	10	-10

用例9

1	8	18
2	1	1
3	4	-4
4	6	6
5	8	-8
6	9	9
7	12	-12
8	14	14
9	17	-17

用例10

1	7	14
2	1	1
3	3	3
4	5	5
5	7	-7
6	9	9
7	11	-11
8	13	13

文章目录

- 华为OD机考:统一考试 C卷 + D卷 + B卷 +A卷
 - 题目描述
 - 输入描述
 - 输出描述
 - 用例1
 - 用例2
 - C++
 - java
 - javaScript
 - python
 - C语言
 - 完整用例
 - 用例1
 - 用例2
 - 用例3
 - 用例4
 - 用例5
 - 用例6
 - 用例7
 - 用例8
 - 用例9
 - 用例10

机考真题 华为OD



CSDN @算法大师