《算法分析与设计》

课后作业

作业编号 作业1

学 号

姓 名

专 业 软件工程

学 院 计算机与人工智能学院

二0二二年三月

第1章 绪 论

1．有一个水平放置的矩形纸箱，现希望用隔板把这个纸箱分割成很多小的格子，每个格子都可以放下若干个物品。从上往下看的效果如下图所示。已知纸箱左上角和右下角的坐标以及每个隔板放入纸箱后的位置坐标。现有若干个物品，已知物品放入纸箱后的位置，求每个格子中有多少个物品？

输入描述：

输入的第一行包含6个整数，分别是n（0<n<=5000），m(0<m<=5000)，x1，y1，x2，y2，分别表示隔板的数量，物品的数量，矩形纸箱左上角点的坐标和右下角点的坐标（从上往下看投影到XY平面的坐标），其后有n行，分别表示隔板的位置，按从左到右的顺序排列，且各个隔板互不相交。每一行有两个整数，Ui和Li，表示第i个隔板的位置坐标在（Ui,y1）和(Li,y2)。其后的m行，每一行有两个整数，Xj 和Yj，分别表示物品的位置。物品不可能正好在隔板上，也不可能在箱子外。

输出描述：

输出n+1行，每一行的形式为：k:h。其中k表示格子的编号，最左边的格子为0，依次递增，最右边的格子为n；h为第k个格子中物品的数量。

样例输入：

5 6 0 10 60 0

3 1

4 3

6 8

10 10

15 30

1 5

2 1

2 8

5 5

40 10

7 9

样例输出：

0: 2

1: 1

2: 1

3: 1

4: 0

5: 1

要求：

1. 用文字描述算法的实现过程。 （20分）
2. 用伪代码描述的实现过程。（20分）
3. 写出算法实现的流程。 （20分）
4. 写出用C/C++语言实现该算法的程序。（40分）

要求以Word文档的形式提交，问题（4）另以C/C++源程序文件提交，将上述两部分内容打包到一个压缩文件中，按下面的要求提交。

评分标准：

1. 算法文字描述（合计20分）
2. 准确的输入/输出变量定义。（5分）
3. 文字表述清楚，有详细的中间变量及计算过程描述，且计算过程正确。（5分）
4. 通过算法的步骤描述能够准确的将输入变量变成输出结果。（10分）
5. 算法伪代码（20分）
6. 准确的输入/输出变量定义。（5分）
7. 伪代码表述规范，有详细的中间变量及计算过程，且计算过程正确。（5分）
8. 通过算法的步骤描述能够准确的将输入变量变成输出结果。（10分）
9. 程序流程（20分）
10. 流程图绘制规范（输入/输出，执行，判断，多选，开始/结束框正确，箭头指向正确）。（5分，错一个扣2分，直到扣完为止）
11. 有明确的输入输出、中间变量定义以及中间变量的计算方法，且计算表达式正确。（5分）
12. 根据流程能得到正确的执行结果，且与算法文字描述一致。（10分）
13. C/C++语言程序实现（20分）
14. 程序编译正确，通过测试数据能得到正确结果。（15分）
15. 程序书写规范，变量函数定义等符合规范，请参照[Google C++ Style Guide](https://google.github.io/styleguide/cppguide.html" \l "Naming)。（10分）
16. 程序与算法描述一致。（10分）
17. 程序中变量定义、函数及主要段落有相应的注释。（5分）

作业提交截止时间：

2022年3月19日20：00，每延后一天扣20分。

作业提交方式：

电子稿，命名规则：学号\_姓名\_作业1.rar

邮件发送给：

* + 1. 该问题的自然语言描述法

输入：隔板的数量n，物品的数量m，矩形纸箱左上角点的坐标(x1,y1)和右下角点的坐标(x2,y2)，隔板的位置Ui和Li，物品的位置Xj 和Yj。

输出：格子的编号i；s[i]为第i个格子中物品的数量。

算法描述：

1. 首先读入n，判断n，不为0时继续运行；
2. 接着依次读入隔板位置数值，再读入物体位置数值，于此同时进行循环判断：
3. 外循环为物体位置，内循环为隔板位置，若读入物体位置X[j],Y[j]满足(L[i]-U[i])\*(Y[j]-y1)-(X[j]-U[i])\*(y2-y1)<0，则判定为物体在该隔板内。格子数组中s[i]数值加1，表明该格子内物品数量加1。循环跳出。
4. i即为格子标号，s[i]即为第i个格子内物品数量。
   * 1. 伪代码描述

输入：n,m,x1,y1,x2,y2,U[i],L[i],X[j],Y[j];

输出：i,s[i];

算法：

**begin**

**while**(scanf("%d",&n),n!=0)

**If** c!=0&&n!=0 **then**

print "\n";

memset(s,0,sizeof(s));

**for**(i<-0;i<n;i++) **do**

**if** (L[i]-U[i])\*(Y[j]-y1)-(X[j]-U[i])\*(y2-y1)<0 **then**

s[i]++;

T<-1;

break;

**if**(t==0)

s[i]++;

**end**

* + 1. 程序流程描述

