

CST2016 3-6 Temperature

题目描述

某气象台每天都要从遍布于各地的观察站采集气温数据，并通过互联网为远程用户提供统计查询服务。其中最常见的一类查询是，根据用户指定矩形区域内所有观察站的观测值计算出平均气温。随着更多观察站的不断建立，原始数据本身的规模急剧膨胀。另外，尽管可以假设每天采集的数据相对固定，但随着用户群体的扩大，查询的频率也日益激增。鉴于传统蛮力算法的效率已无法满足实用要求，气象台只好请你帮忙，通过改进数据结构和算法，提高查询的效率。

借助气象台提供的一组函数接口，服务器端可访问已采集到的所有数据，并报告查询结果。

接口说明

```
int GetNumOfStation(void);
```

该函数必须 首先 调用，返回现有观察站的总数n。

```
void GetStationInfo(int no, int *x, int *y, int *temp);
```

获得第no个 ($0 \leq no < n$) 观察站的信息：其地理坐标(*x,*y)及其所测温度值*temp。各观测站的测量精度统一以0.01°C为基准单位，比如12.34°C表示为 整数 1234。

```
int GetQuery(int *x1, int *y1, int *x2, int *y2);
```

接收下一查询请求。返回值1对应于一次有效的查询。矩阵区域的四边分别与x或y轴平行，(*x1,*y1)和(*x2,*y2)描述其一条对角线。恰好被矩形边界穿过的观察站，也视作落在其中。若返回0，则表示没有更多的查询，你的程序可以退出。

```
void Response(int temp);
```

针对当前的查询，在计算出对应的平均气温后，你可通过这一接口报告所得数值(截断取整，比如12.345°C输出为1234，-12.345°C输出为-1234)。

特别注意：每调用GetQuery()接收一次查询后，若未能通过Response()函数报告该次查询的结果就再次调用GetQuery()接收下一查询，则将因为前次查询的结果无法报告而注定输出错误。也就是说，GetQuery()和Response()必须 交替 调用，各n次。

测试说明

为便于你调试和测试，随题还附带有temperature.h和temperature_lib.c文件。前者约定了上述接口，后者是这组接口的一种实现——OJ上的实现与之不同，但接口完全一致。调试时可将它们与你的代码一同编译，但在线测试时不必提交；即便提交，OJ也会自动忽略它们。

下载接口文件 (attachment/60fe/60fec26fef5ccada8e4e8845b808985ad9080785.zip)

输入

脱机调试时，temperature_lib.c所实现的三个输入接口，实际上是从当前目录下的temperature.in文件读入数据，因此通过按如下格式更改该文件，即可设定不同的输入数据：

第一行为两个整数：观察站总数n，所需查询的总次数m。
以下n行分别描述各观察站：位置坐标为整数(x, y)，该站所测得温度值为整数t。
再以下m行分别对应于各次查询操作，用整数(x1, y1)和(x2, y2)描述其一条对角线。

输出

脱机调试时，temperature_lib.c所实现的Response()接口会在程序运行后，将所有的输出结果写入temperature.out文件。

文件共m行，各含1个整数，表示每次查询所得平均温度。

若查询区域不含任何观测站，则输出0。

输入样例

```
4 2
0 0 1000
1 1 1300
2 2 1600
3 3 1100
0 0 1 1
0 0 10 10
```

输出样例

```
1150
1250
```

数据范围

$1 \leq n \leq 50,000$

$1 \leq m \leq 500,000$

观测站坐标取值范围是 $[-2^{31}, 2^{31})$

资源限制

时间限制：5 sec

内存限制：256 MB

提示

- 温度计算请使用"64位整数", 以保证累加不致溢出
- range tree

UI powered by Twitter Bootstrap (<http://getbootstrap.com/>).
Tsinghua Online Judge is designed and coded by Li Ruizhe.
For all suggestions and bug reports, contact [oj\[at\]liruizhe\[dot\]org](mailto:oj[at]liruizhe[dot]org).