My Courses

CST2020 2-5 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，用户指定一个矩形区域，查询该区域内的最高气温和最低气温。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。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

我们提供了一个交互库，包括一个头文件 temperature.h 和一个库文件 temperature_lib.cpp。头文件包含了初始化函数和查询函数的声明，库文件包含了气象台使用这些接口的主函数的实现。

编译时，我们会将库文件和你的程序一起编译和链接，因此**你只需在你的程序中包含头文件，并实现初始化函数和查询函数**。

```
void init(int n, const int *x, const int *y, const int *temp);
```

这是初始化函数。交互库首先会调用这个函数，你可以在此对你的程序进行初始化。

其中 n 表示观察站的总数，x, y, temp 三个数组，分别表示每个观察站的地理坐标 (x[i], y[i]) 以及其所测温度值 temp[i]（0 <= i < n）。

```
void query(int x1, int x2, int y1, int y2, int *tmin, int *tmax);
```

这是查询函数。在初始化完成后，交互库会调用这个函数 m 次，每一次你都需要进行相应地查询并报告结果。

查询的矩形区域的四边分别与 x 或 y 轴平行，(x1, y1) 和 (x2, y2) 描述其一条对角线。恰好位于矩形边界的观察站，也视作落在其中。

对于每次查询，你需要报告区域内的最低气温、最高气温，分别存入 *tmin、*tmax。如果在区域内没有观察站，请报告 -1 -1。

测试说明

我们的交互库的具体实现如下：

首先从标准输入读取数据。我们提供的交互库和 OJ 上评测用的交互库的读入方式可能并不相同，但是这不会影响你做题。

然后调用 init 函数。

接着调用若干次 query 函数，并将 query 函数的结果写入标准输出。

如果你需要在本地测试你的程序，你可以使用 g++ your_code.cpp temperature_lib.cpp -std=c++14 -O2 -Wall -o temperature.exe 来编译，再运行 temperature.exe 来测试。

你需要严格按照输入格式来向交互库输入数据，否则不能保证交互库正常运行。输入结束后，在 cmd 或 PowerShell 里，可以用 Ctrl + Z 表示 EOF；在 bash 里，可以用 Ctrl + D 表示 EOF。

程序运行结束后，你可以将标准输出中的内容与答案进行比较，来知道你的输出是否正确。

我们还提供了一个样例程序 temperature_sample.cpp，实现了“每次查询都报告随机数”的算法。你可以参考该程序来理解交互库和解题，或者不参考，这将与你的得分无关。

请注意，你只需提交实现了 init、query 函数的源程序文件（例如提交单文件 temperature_sample.cpp）。我们会自动加入 temperature.h、 temperature_lib.cpp，请不要提交这两个文件，否则可能会导致编译错误。

交互库和运行库会使用一定时间、空间，且这部分空间会计入评测结果。我们保证交互库消耗的时间不超过 0.1 sec，交互库和运行库消耗的空间不超过 (12 * n) 字节 + 14 MB。

下载交互库和样例程序 (attachment/c903/c903208bb02fc2882b9e3899be1962cbdf1c488c.zip)

输入

对于我们提供的交互库，输入格式如下：

第一行为 1 个整数，表示观察站个数 n。

接下来 n 行每行 3 个整数，表示一个观测站的横坐标、纵坐标、气温。

接下来有 m 行，每行 4 个整数，表示一次查询区域 x1、x2、y1、y2。

输出

对于我们提供的交互库，输出格式如下：

共 m 行，每行 2 个整数，表示最低气温和最高气温，对应一次查询结果。

输入样例

```
4
0 0 100
1 2 200
1 2 3000
3 3 4000
0 0 0 0
-1 -1 -1 -1
0 2 0 2
1 3 2 3
```

这是对于我们提供的交互库的一组输入样例，可用于本地调试。在 OJ 上评测时，由于交互库实现不同，输入格式不一定是这样；请交给交互库处理，不要自己操作 stdin。

* 该样例是第一个测试点

输出样例

```
100 100
-1 -1
100 3000
200 4000
```

这是该组输入样例的标准答案，可用于本地测试。在 OJ 上评测时，由于交互库实现不同，输出格式不一定是这样。

限制

禁止覆盖交互库提供的主函数。禁止操作标准输入流、标准输出流。

init 函数传入的 x、y、t 必须由交互库释放，禁止在 init 函数之外使用这片空间。

数据范围

1 <= n <= 200,000，1 <= m <= 160,000

气温为整数，范围是 [0, 2^31)

观测站坐标为整数，范围是 [-2^31, 2^31)；可能有坐标重合的观测站

对于每次查询，保证 x1 <= x2，y1 <= y2

资源限制

时间：2.5 sec

空间：256 MB

提示

- Multi-Level Search Tree -> 2D Range Query（讲义 09-C）
  - 为了避免遍历区域内所有观测站，可在 y-Query 各内部节点记录整棵子树的最低气温、最高气温。
  - 为了节省空间，需精简数据结构，可参考完全二叉树基于向量的紧凑表示（讲义 12-B1 第一页）。
  - 不必实现 Fractional Cascading（讲义 09-XA）也能通过本题。
- 2d-树（讲义 09-B2）
  - 为了避免遍历区域内所有观测站，可在各内部节点记录整棵子树的最低气温、最高气温。
  - 内存连续性对时间有较大影响，尽量将同一子树的节点放在一起。
