

NOI 2017 全国青少年信息学奥林匹克竞赛

重庆代表队选拔赛 第二试

试题一览

题目名称	老 C 的任务	老 C 的方块	老 C 的键盘
题目英文名	task	block	keyboard
输入文件名	task.in	block.in	keyboard.in
输出文件名	task.out	block.out	keyboard.out
测试点数	10	10	10
每个测试点分值	10	10	10
满分分值	100	100	100
时间限制	2 秒	1 秒	2 秒
内存限制	512MB	512MB	512MB

2017 年 4 月 9 日

5 小时完成

注：允许使用 64 位整数和 STL，但这不一定是解题必须的

老 C 的任务

(task.c/cpp/pas)

【问题描述】

老 C 是个程序员。

最近老 C 从老板那里接到了一个任务——给城市中的手机基站写个管理系统。作为经验丰富的程序员，老 C 轻松地完成了系统的大部分功能，并把其中一个功能交给你来实现。

由于一个基站的面积相对于整个城市面积来说非常的小，因此每个的基站都可以看作坐标系中的一个点，其位置可以用坐标 (x, y) 来表示。此外，每个基站还有很多属性，例如高度、功率等。运营商经常会划定一个区域，并查询区域中所有基站的信息。

现在你需要实现的功能就是，对于一个给定的矩形区域，回答该区域中（包括区域边界上的）所有基站的功率总和。如果区域中没有任何基站，则回答 0。

【输入格式】

输入文件 task.in。

第一行两个整数 n, m ，表示一共有 n 个基站和 m 次查询。

接下来一共有 n 行，每行由 x_i, y_i, p_i 三个空格隔开的整数构成，表示一个基站的坐标 (x_i, y_i) 和功率 p_i 。不会有两个基站位于同一坐标。

接下来一共有 m 行，每行由 $x1_j, y1_j, x2_j, y2_j$ 四个空格隔开的整数构成，表示一次查询的矩形区域。该矩形对角坐标为 $(x1_j, y1_j)$ 和 $(x2_j, y2_j)$ ，且 4 边与坐标轴平行。

【输出格式】

输出文件 task.out。

输出 m 行，每行一个整数，对应每次查询的结果。

【输入输出样例 1】

task.in	task.out
4 2	11
0 0 1	4
0 1 2	
2 2 4	
1 0 8	
0 0 1 1	
1 1 5 6	

见选手目录下的 task/task1.in 与 task/task1.ans。

【输入输出样例 2】

task.in	task.out
3 2	0
-100 0 16	112
1 -10 32	
1000 100 64	
0 0 0 1	
-1000 -1000 10000 10000	

见选手目录下的 task/task2.in 与 task/task2.ans。

【数据规模与约定】

对于第 1~2 个测试点, $1 \leq n, m \leq 100$;

对于第 3~5 个测试点, $1 \leq n \leq 50000, 1 \leq m \leq 10000$;

对于第 6~10 个测试点, $1 \leq n \leq 100000, 1 \leq m \leq 100000$, 数据有梯度;

对于所有测试点, $-2^{31} \leq x_i, y_i, p_i, x_{1j}, y_{1j}, x_{2j}, y_{2j} < 2^{31}, x_{1j} \leq x_{2j}, y_{1j} \leq y_{2j}$ 。

老 C 的方块

(block.c/cpp/pas)

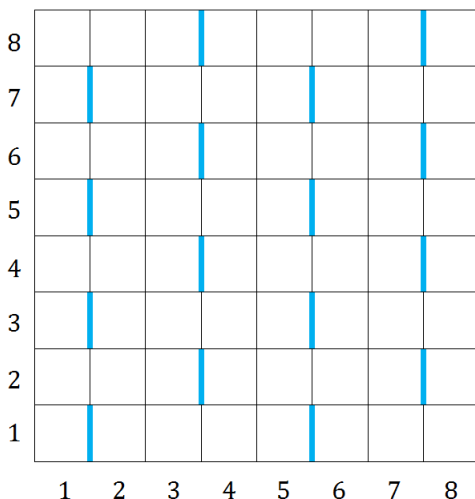
【问题描述】

老 C 是个程序员。

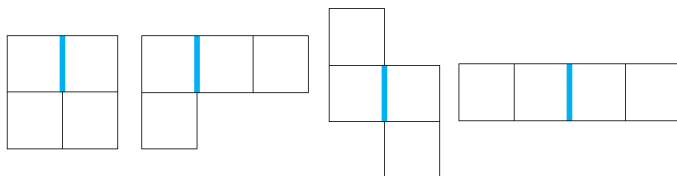
作为一个懒惰的程序员，老 C 经常在电脑上玩方块游戏消磨时间。

游戏被限定在一个由小方格排成的 R 行 C 列网格上，如果两个小方格有公共的边，就称它们是相邻的，而且有些相邻的小方格之间的公共边比较特殊。特殊的公共边排列得有很强的规律。首先规定，第1行的前两个小方格之间的边是特殊边。然后，特殊边在水平方向上每4个小方格为一个周期，在垂直方向上每2个小方格为一个周期。所有的奇数列与下一列之间都有特殊边，且所在行的编号从左到右奇偶交替。

下图所示是一个 $R = C = 8$ 的网格，蓝色标注的边是特殊边。首先，在第1行，第1列和第2列之间有一条特殊边。因为垂直方向周期为2，所以所有的奇数列，第1列和第2列之间都有特殊边。因为水平方向周期为4，所以所有奇数列的第5列和第6列之间也有特殊边，如果网格足够大，所有奇数列的第9列和第10列、第13列和第14列之间都有特殊边。因为所有的奇数列和下一列之间都有特殊边，所以第3列和第4列、第7列和第8列之间也有特殊边，而所在行的编号从左到右奇偶交替，所以它们的特殊边在偶数行。如果网格的规模更大，我们可以用同样的方法找出所有的特殊边。



网格的每个小方格刚好可以放入一个小方块，在游戏的一开始，有些小方格已经放上了小方块，另外的小方格没有放。老 C 很讨厌下图所示的图形，如果他发现有一些小方块排列成了它讨厌的形状（特殊边的位置也要如图中所示），就很容易弃疗，即使是经过任意次旋转、翻转后排列成讨厌的形状，老 C 也同样容易弃疗。



为了防止弃疗，老 C 决定趁自己还没有弃疗，赶紧移除一些格子里小方块，使得剩下的小方块不能构成它讨厌的形状。但是游戏里每移除一个方块都是要花费一些金币的，每个方块需要花费的金币有多有少参差不齐。老 C 当然希望尽可能少的使用游戏里的金币，但是最少要花费多少金币呢？老 C 懒得思考，就把这个问题交给你了。

【输入格式】

输入文件 `block.in`。

第一行有3个正整数 C, R, n ，表示 C 列 R 行的网格中，有 n 个小方格放了小方块。

接下来 n 行，每行3个正整数 x, y, w ，表示在第 x 列第 y 行的小方格里放了小方块，移除它需要花费 w 个金币。保证不会重复，且都在网格范围内。

【输出格式】

输出文件 `block.out`。

输出一行，包含一个整数，表示最少花费的金币数量。

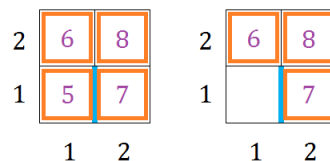
【输入输出样例 1】

<code>block.in</code>	<code>block.out</code>
2 2 4 1 1 5 1 2 6 2 1 7 2 2 8	5

见选手目录下的 `block/block1.in` 与 `block/block1.ans`。

【输入输出样例 1 说明】

如图所示， 2×2 的网格里，每个小方格都放了小方块，移除需要花费的金币数量如图中标注所示。这4个小方块恰好构成了一个老 C 讨厌的图形，将第1列第1行的小方块移除，花费5个金币，剩下的方块不能形成老 C 讨厌的图形，显然这种移除方案花费的金币最少。



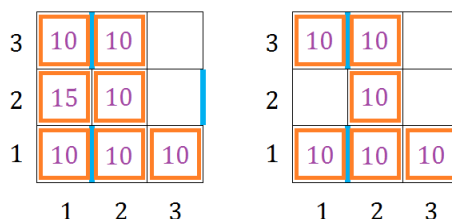
【输入输出样例 2】

<code>block.in</code>	<code>block.out</code>
3 3 7 1 1 10 1 2 15 1 3 10 2 1 10 2 2 10 2 3 10 3 1 10	15

见选手目录下的 `block/block2.in` 与 `block/block2.ans`。

【输入输出样例 2 说明】

如图所示。容易发现，如果不移除第1列第2行的小方块，则至少要移除两个小方块，才能不包含老 C 讨厌的图形，花费至少20个金币；而删除第1列第2行的小方块后，原有的讨厌图形全都不存在了，只需要花费15个金币。



【数据规模与约定】

对于第 1~2 个测试点, $1 \leq C, R \leq 100, 1 \leq n \leq 20$;

对于第 3~6 个测试点, $1 \leq C, R \leq 10^5, 2000 \leq n \leq 5000$, 数据有梯度;

对于第 7~10 个测试点, $1 \leq C, R \leq 10^5, 30000 \leq n \leq 10^5$, 数据有梯度;

对于所有测试点, $1 \leq C, R, n \leq 10^5, 1 \leq w \leq 10^4$ 。

老 C 的键盘

(keyboard.c/cpp/pas)

【问题描述】

老 C 是个程序员。

作为一个优秀的程序员，老 C 拥有一个别具一格的键盘，据说这样可以大幅提升写程序的速度，还能让写出来的程序在某种神奇力量的驱使之下跑得非常快。

小 Q 也是一个程序员。有一天他悄悄潜入了老 C 的家中，想要看看这个键盘究竟有何妙处。他发现，这个键盘共有 n 个按键，这 n 个按键虽然整齐的排成一列，但是每个键的高度却互不相同。聪明的小 Q 马上将每个键的高度用 $1 \sim n$ 的整数表示了出来，得到一个 $1 \sim n$ 的排列 h_1, h_2, \dots, h_n 。为了回去之后可以仿造一个新键盘（新键盘每个键的高度也是一个 $1 \sim n$ 的排列），又不要和老 C 的键盘完全一样，小 Q 决定记录下若干对按键的高度关系。作为一个程序员，小 Q 当然不会随便选几对就记下来，而是选了非常有规律的一些按键对：对于 $i = 2, 3, \dots, n$ ，小 Q 都记录下了一个字符 $<$ 或者 $>$ ，表示 $h_{\lfloor i/2 \rfloor} < h_i$ 或者 $h_{\lfloor i/2 \rfloor} > h_i$ 。于是，小 Q 得到了一个长度为 $n - 1$ 的字符串，开开心心的回家了。

现在，小 Q 想知道满足他所记录的高度关系的键盘有多少个。虽然小 Q 不希望自己的键盘和老 C 的完全相同，但是完全相同也算一个满足要求的键盘。答案可能很大，你只需要告诉小 Q 答案 mod 1,000,000,007 之后的结果即可。

【输入格式】

输入文件名为 keyboard.in。

输入共 1 行，包含一个正整数 n 和一个长度为 $n - 1$ 的只包含 $<$ 和 $>$ 的字符串，分别表示键盘上按键的数量，和小 Q 记录的信息，整数和字符串之间有一个空格间隔。

【输出格式】

输出文件名为 keyboard.out。

输出共 1 行，包含一个整数，表示答案 mod 1,000,000,007 后的结果。

【输入输出样例 1】

keyboard.in	keyboard.out
5 <>><	3

见选手目录下的 keyboard/keyboard1.in 与 keyboard/keyboard1.ans。

【输入输出样例 1 说明】

共 5 个按键，第 1 个按键比第 2 个按键矮，第 1 个按键比第 3 个按键高，第 2 个按键比第 4 个按键高，第 2 个按键比第 5 个按键矮。

这 5 个按键的高度排列可以是 (2,4,1,3,5), (3,4,1,2,5), (3,4,2,1,5)。

【输入输出样例 2】

keyboard.in	keyboard.out
5 <<<<	8

见选手目录下的 keyboard/keyboard2.in 与 keyboard/keyboard2.ans。

【输入输出样例 2 说明】

这5个按键的高度排列可为(1,2,3,4,5), (1,2,3,5,4), (1,2,4,3,5), (1,2,4,5,3), (1,2,5,3,4), (1,2,5,4,3), (1,3,2,4,5), (1,3,2,5,4)。

【输入输出样例 3】

keyboard.in	keyboard.out
5 <<>>	18

见选手目录下的 keyboard/keyboard3.in 与 keyboard/keyboard3.ans。

【数据规模与约定】

测试点编号	n	其他说明
1	10	无
2	18	
3	20	
4	25	
5	30	
6	80	
7	100	小 Q 记录的字符串中只包含<符号
8	100	小 Q 记录的字符串中只有最后一个字符是>
9	100	无
10	100	