

1 Problem 1. Distance

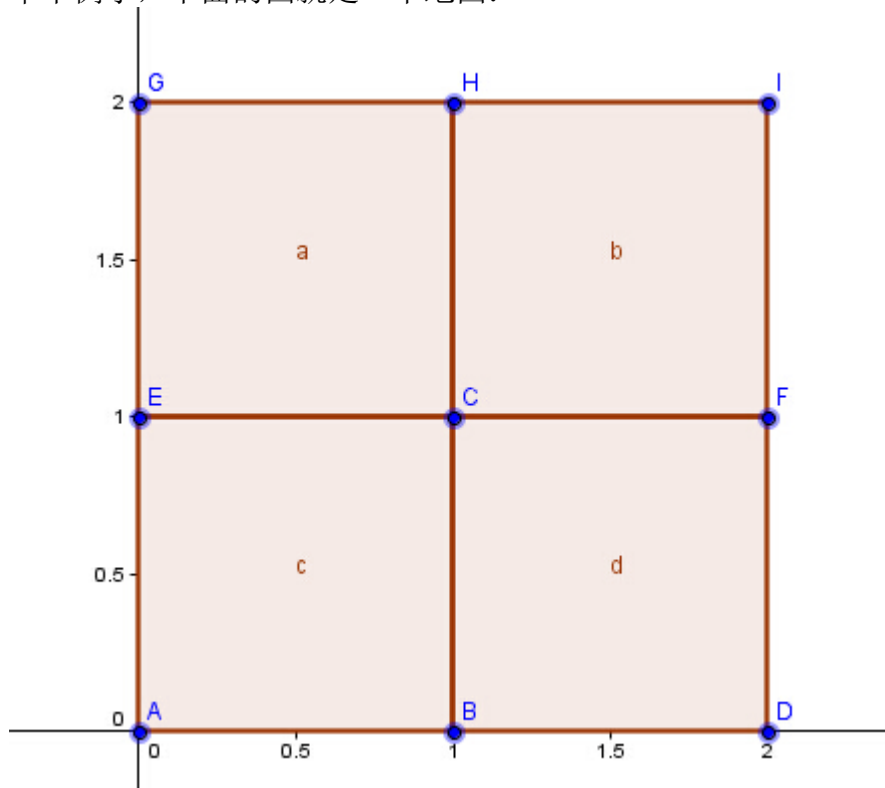
1.1 题目描述

File Name : distance.cpp/c/pas

小L和小G是一对好朋友，他们居住在一个美好的世界里，打开地图，你能看到世界由一个个紧挨着的国家构成，我们可以用一个平面图来表示。小L和小G都很喜欢旅游，一下子到这个国家一下子到那个国家，所以他们经常是分开的。但是有时候，小L想小G了，他就会去找她玩，从一个国度到另一个国度是要收费的。

对于地图，首先输入 n ($n \leq 10^5$)个点 (x_i, y_i) ，然后再输入 m ($m \leq 10^5$)条边 (u, v, w) ，表示在点 (x_u, y_u) 和 (x_v, y_v) 之间连接一条线段，线段的权值为 w ，保证所有线段仅在端点处相交，而且最后生成的平面图是连通的，每个国家一定是一个封闭的多边形（里面可能夹杂了一些线段），更严谨地我们可以说是平面图的一个域。

举个例子，下面的图就是一个地图：



这个图中一共有四个国家：小正方形a,b,c,d。还有一个无穷大的区域（就是除了阴影部分之外的区域）。

现在有 $Q(Q \leq 100000)$ 组询问，对于第 i 组询问给定两个点的坐标 (a, b) 和 (c, d) ，分别代表小L的现在位置，和小G 的现在位置。

首先我们定义一下路径，路径可以是任意以点 (a, b) 出发到点 (c, d) 结束的有向曲线段，一条路径可能按顺序割断了若干条图中的线段（就是 m 条边中的某一条），一条边可以被经过若干次。

因为代价的计算仅仅和边有关，所以我们可以用一个边的序列来表示一条路径 $\langle e_1, e_2, \dots, e_k \rangle$ ，其中路径中可能出现相同的边。

令 w_i 表示路径上边 e_i 的权值，那么这条边对总代价的贡献就是 $f(i)w(i)$ ，其中

$$f(i) = f(i-1) * (c^i - 1) \mod P \quad (1)$$

其中 $c = 4263, P = 1632899, f(0) = 1$ 。

最后的 $\sum_{i=1}^k f(i)w(i)$ 就是这条路径的代价，然后我们要询问的是小L从现在的位置出发，至少割过 $t(100 \leq t \leq 1000)$ 次边到达小G的位置的所有路径中代价最小的那条路径的代价是多少。

注意我们把那个无穷大的区域也看成是一个国家，也就是小L和小G都有可能出现在那个区域。

1.2 输入格式

Input File : distance.in

第一行两个正整数 n, m ，分别代表平面图一共有 n 个点和 m 条边。

接下来 n 行，每行两个非负整数 (x, y) 表示第 i 个点的坐标（从1开始标号）

再接下来一共 m 行，每行三个正整数 (u, v, w) ，表示第 u 个点和第 v 个点之间有一条权值为 w 的边。

然后是单独一行一个正整数 Q ，表示有 Q 组询问。

接下来每行四个非负实数 a, b, c, d 和一个正整数 t ，代表小L的位置在 (a, b) ，小G的位置在 (c, d) ，然后路径要求至少走 t 步。

1.3 输出格式

Output File : distance.out

共 Q 行，每行一个正整数代表对每个询问的答案。

1.4 数据范围

$1 \leq n, m \leq 10^5, 0 \leq x, y \leq 10^9, 1 \leq w \leq 500, 10^2 \leq t \leq 10^3$

$0 \leq a, b, c, d \leq 10^9$ 且都是0.5的整数倍, 保证任意询问点都不出现在线段上。

50%的数据满足 $0 \leq x, y \leq 10^3$, 且所有线段长度为1。

时间限制: 3s。内存限制: 512MB。

2 Problem 2. Matrix

2.1 题目描述

File Name : matrix.cpp/c/pas

小L和小G是一对好朋友, 小G很喜欢数学, 尤其是矩阵, 有一天她弄了10种大小为 10×10 矩阵 $T[1] \dots T[10]$ 。然后随机了一个序列 $a_1 \dots a_n$ ($1 \leq a_i \leq 10$), 她要小L算出 $T[a_1] \times T[a_2] \dots T[a_n]$ 的结果 (一个矩阵)。

小L觉得小G是在刁难他, 于是他就想看看小G是不是真的自己会计算, 于是他将矩阵 $T[a_1], T[a_2] \dots T[a_n]$ 按顺序从左到右写成一排。

然后按顺序执行100次操作, 对于第 i 次操作, 小L随机产生一个 $[1..n+i-1]$ 之间的正整数 x , 和一个矩阵 G , 然后他就在序列中第 x 个矩阵的后面紧挨着插入一个新矩阵 G , 然后序列就多了一个矩阵。

然后他把这个新序列 (包括 $n+100$ 个矩阵) 给小G去算, 因为插入100个新矩阵规模没有什么变化, 所以这样对小G是公平的, 但是小G最近事情很多挺忙的, 她又不想让小L生气, 所以她找你来帮忙应付一下小L。

你需要告诉小L这个新矩阵序列从左到右的乘积, 注意所有数字运算是在模 10^9+7 意义下进行的。

2.2 输入格式

Input File : matrix.in

首先是按顺序输入 $T[1] \dots T[10]$ (注意对于所有测试数据这10个矩阵都不会有变化, 我们会放到TheMatrix.txt中)

对于每个 10×10 的矩阵, 我们用10行来输入, 每行空格隔开10个整数, 相邻两个矩阵之间空一行。

接下来单独一行一个数字代表 n 。

然后下一行有空格隔开的 n 个整数表示 a 序列。

接下来有100个部分，每个部分描述了一个要插入的矩阵，首先是单独一行输入 x ，然后用10行来输入该要被插入的矩阵。

2.3 输出格式

Output File : matrix.out

共10行，每行10个空格隔开的整数，表示答案矩阵。

2.4 数据范围

对于 $T[1..10]$ 每个矩阵中的每个数字的大小都是 $[0, 10^9 + 6]$ 之间的整数。

对于后来插入的每个矩阵中的每个数字大小都是 $[1..10]$ 之间的整数。

30%的数据满足 $1 \leq n \leq 1000$

100%的数据满足 $1 \leq n \leq 10^6$

时间限制：1s。内存限制：256MB。

3 Problem 3. Circle

3.1 题目描述

File Name : Circle.cpp/c/pas

小L和小G是一对好朋友。有一天他们来到了游乐园，看到一个非常有意思的套圈圈游戏，一共有 n 个奖品放在地面上，我们给他**随机标号**1... n ，对于第 i 个奖品可以点 (x_i, y_i) 来表示它的位置。

小L为了在小G面前留一个很好的印象，就跟负责的人说，你随便指定一个区间 $[L, R]$ ，然后给我一个半径为 r 的圈圈，我保证在800里的距离之外用这个圈圈套中所有标号在 $[L, R]$ 中的奖品（可能还多套了别的），而且你们都不能找到一个更小的圈圈来做到跟我一样的事情。

而且为了让他们服气，小L说你们可以考我 m 次，每次就指定一个区间 $[L_i, R_i]$ ，然后小L就告诉他们我要一个半径为 r_i 的圈圈，之后就开始表演800里外一圈套中奖品。

小G才不感兴趣呢，她满脑子都想着她的矩阵，但是他记下了小L说的每次圈圈的大小的**整数部分**为 r_i ，对于第 i 次套圈套完之后，她能马上构造一个大小为 $i \times i$ 的矩阵 F_i

$$F_i = \begin{pmatrix} r_1^0 & r_1^1 & \dots & r_1^{i-1} \\ r_2^0 & \dots & \dots & r_2^{i-1} \\ \dots & \dots & \dots & \dots \\ r_i^0 & \dots & \dots & r_i^{i-1} \end{pmatrix} \quad (2)$$

然后口算它的行列式模 $10^9 + 7$ 的结果，并把它加上 $L_i \times R_i$ 得到 ans_i ，小声地自言自语嘀咕出这个数字。

回家之后呢，小G说小L一点都不关心她，只管自己套圈呢，小L听到这话一下子紧张了，不过一会儿之后，他对小G说，哪里有，我很关心你的好不好，每次套圈完之后的你说的数字我都记得呢。小G听到这个话之后一下子就高兴了，真的丫，那我每次都说了写什么呢？

这个嘛，小L哪里会记得，他赶紧偷偷发消息给套圈工作人员，要到了他们每个奖品的位置，每次询问的区间。

小L暗中联系了正在敲代码的你，要你帮忙写个小代码来算一下小G每次说的数字，作为回报，他能让你在考场上多拿100分。

3.2 输入格式

Input File : circle.in

第一行一个正整数 n ，代表有 n 个奖品。

接下来 n 行，每行两个非负整数 (x_i, y_i) ，表示标号为 i 的奖品的坐标。

然后输入一个数 m ，表示一共有 m 个询问区间，接下来 m 行，每行两个正整数 $[L_i, R_i]$ 。

3.3 输出格式

Output File : circle.out

共 m 行，表示每个询问的答案。

3.4 数据范围

数据保证 $1 \leq n, m \leq 10^5, 0 \leq x_i, y_i \leq 10^4$ 。所有坐标都是随机生成。

询问序列的生成方法：首先随机生成 $2 * 10^5$ 个区间 $(a[i], b[i])$ 。然后对于第 i 个询问，我们随机一个 $[1, 2 * 10^5]$ 的整数 x ，令 $L[i] = a[x], R[i] = b[x]$ 。

对于30%的数据，满足 $n \leq 10^2, m = 1$ 。

时间限制：2s。内存限制：256MB。