

NOI模拟试题

(请选手务必仔细阅读本页内容)

题目概览

中文题目名称	那些年，我们一起参加的艺术考试	铁拳	Onepiece
英文题目名称	Art	Tekken	Onepiece
可执行文件名	Art.exe	Tekken.exe	Onepiece.exe
输入文件名	Art.in	Tekken.in	Onepiece.in
输出文件名	Art.out	Tekken.out	Onepiece.out
运行时间上限	1 秒	4 秒	1 秒
运行内存上限	256M	256M	256M
比较方式	全文比较	全文比较	全文比较
题目类型	传统	传统	传统

注意事项：

1. 请用**小写**命名文件名，每题交源文件，子目录也放。
2. C/C++中函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. C++选手**禁用**所有以“__”开头的库函数，但**开放** STL。
4. 所有试题均**不打开**编译优化开关。

那些年，我们一起参加的艺术考试

(art. pas/c/cpp)

【问题描述】

又到一年圣诞节，我们一起来参加一年一度的艺术考试。

作为一个 N 年没上美术课的理科生，Wayne 在考场上手一抖，把调好的颜料给洒了（我擦 ==），纸上一片凄惨……他可不想让自己的成绩也变得如此凄惨，从而决定力挽狂澜！

已知这次考试的形式如下：

给出一张等分成 $n * m$ 个正方形小格的白色方格纸，每个小格只会有一种颜色，初始为白色。

现在一些方格被染上颜色，而 Wayne 可以用一些棉球来对方格重新染色，这些棉球只能把方格染成它的颜色。现在某些格子上会包含一个棉球，只要 Wayne 不操作，方格就不会被棉球染色。

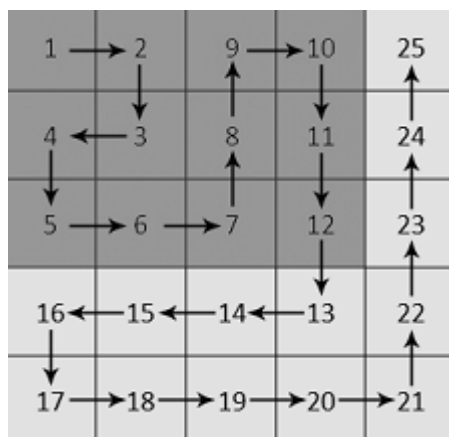
要想搞定这个烂摊子，Wayne 必须对方格纸重新染色。为了更好地说明操作过程，我们引进一个队列，里面按顺序存放棉球，初始为空。

一开始，Wayne 把一个棉球拿走，放到队列尾。只要队列中还有棉球，我们就重复以下操作：

从队列头拿出一个棉球，设它的颜色为 c ，它所处方格现在的颜色为 c' 。如果 c 和 c' 不相同 并且 c' 不是白色，那么 Wayne 就把所有颜色为 c' 的方格按一定顺序重新染色为 c ，当经过一个方格时，把其上的棉球拿走放到队列尾。一个棉球最多只能被拿走一次。

对于染色顺序的定义：

设棉球原来所处的坐标为 (x, y) ，令 $x' = \max(x, n-x+1)$ ， $y' = \max(y, m-y+1)$ ，原图的 (x, y) 就对应下图的 (x', y') 。那么对于原图一个格子 (a, b) ，它在新图中的坐标则为 $(a-x+x', b-y+y')$ 。



比如说， $n=m=5$ ，棉球的原图坐标为 $(3,2)$ ，那么它在上图中的坐标则为 $(3,4)$ 。那么原图中一个坐标为 $(4,3)$ 的待染色格子，在新图中坐标则为 $(4,5)$ ，权值则为 22。把所有要重新染色的格子的权值从小到大排序，则权值小的格子先染色。

现在 Wayne 告诉你方格纸上的初始颜色，每个棉球的具体位置 以及 一开

始被拿走的棉球是哪个，请你告诉 Wayne 重新染色次数一共是多少。

【输入格式】

第一行两个正整数 n 和 m ($1 \leq n, m \leq 300$)，分别表示方格纸的高度和宽度。

接下来 n 行，每行 m 个整数，表示每个格子的初始颜色。

再接着 n 行，每行 m 个整数，表示格子下方棉球的颜色， -1 表示没有棉球。

颜色的范围是 $[0, 1e9]$ ， 0 表示白色。

最后一行包含两个正整数 x 和 y ，表示一开始被拿走棉球的坐标。

坐标从 1 开始计数，从上到下，从左到右依次递增。

数据保证合法。

【输出格式】

一行一个整数，表示重新染色的次数。

【输入输出样例】

art.in	art.out
3 4 1 0 1 0 0 0 0 0 0 1 0 0 0 -1 1 -1 -1 -1 -1 -1 -1 1 0 0 0 0 0 0 0 0 -1 -1 3 2	6

【样例说明】

一开始颜色为 $1e9$ 的棉球被拿走，把 3 个颜色为 1 的格子染成 $1e9$ ，并且依次把颜色为 0 和 1 的棉球放到队列里面。

接着处理颜色为 0 的棉球，把三个颜色为 $1e9$ 的格子染成 0 。

接着处理颜色为 1 的棉球，由于它原来所处格子的颜色已经被染成了白色，所以不作任何处理。

这样一共进行了 6 次重新染色。

【数据规模】

对于 40% 的数据， $n, m \leq 10$ ；

对于 60% 的数据，颜色范围为 $[0, 1e6]$ ；

对于 100% 的数据， $n, m \leq 300$ ，颜色范围为 $[0, 1e9]$ 。

铁拳

(tekken.pas/c/cpp)

【问题描述】

经过了可怕的第三次世界大战后，国家政府崩溃，各大财团趁机夺取掌控世界。长年战争后，八大财团幸存并割据一方，其中最强的当属掌控北美的铁拳。

在铁拳财团所维护的文明区域中，有一项最为光荣、重要的赛事——Iron Fist，也就是铁拳大赛。IF 中云集了世界各地各财团鼎力资助的世外高手，只为了赢得 IF Champion，得到无上的荣耀，当然还有随之而来的权力。本来一切秩序井然，但一个来自贫民窟的少年风间仁意外地在海选中赢了 IF 正式选手，获得了决赛资格，从此格局被打乱……

为了应对这突如其来的变数，IF 管理层决定先对联盟中所有的选手进行评估，以更好地掌握大局。

已知最近 m 届比赛出现过的 n 位选手，背后都有着各自财团的资助，并且签下了合同。由于这是各财团的高度机密，合同的具体细节无从得知，但铁拳财团的间谍们通过各种渠道得知了每个选手的薪金范围（显然薪金是非负数）。

对于最近 m 届的 IF 比赛（从 1 开始编号），每一届联盟都会进行清算，通过国际金融手段准确计算出这一届联盟选手身价总和的变化。每一届中，会有一些新选手加入，也会有部分选手在比赛中丧失了战斗能力，而被踢出联盟，流放到贫民窟。

现在给出联盟中 n 位选手的身价范围，以及他们 进入联盟的届数（0 表示在 $m+1$ 届以前就已经是联盟选手）和 离开联盟的届数（0 表示是现役选手）。同时给出最近 m 届中，每一届联盟选手身价总和减去上一届的值。

请你根据现有信息，尽可能准确地给出每个选手可能的薪金范围。各选手之间的薪金范围可以不同时成立，但对于一位选手的范围中的每一个数，都必须至少存在一种合法方案使该选手能得到相应薪金，而且这个范围跨度要尽可能大。

如果输入信息有误，请输出 -1，表示无解。

【输入格式】

第一行一个正整数 m ，意义见上（下同）。

第二行包含 m 个整数，第 i 个表示第 i 届中 选手身价总和 的变化情况。

第三行一个正整数 n 。

接下来 n 行，每行包含四个整数，分别表示 身价下限 、 身价上限 、 出道届数 、 退役届数，细节请参照上文。

保证出道时间严格比退役时间小（0 除外）。

【输出格式】

如果输入信息有误，仅输出一行一个整数 “-1”（不含双引号）。

否则输出 n 行，每行两个实数，第 i 行表示第 i 个选手实际身价的准确范围。

保留两位小数，需要与标准答案完全相同才得分。

【输入输出样例 1】

tekken.in	tekken.out
2	1.00 2.00
5 -1	2.00 3.00
3	1.00 1.00
1 4 1 0	
2 3 1 0	
1 5 1 2	

【样例说明 1】

第二届只有 3 号离开了，可以锁定 3 号的薪金是 1。

如此一来，1 号和 2 号薪金之和为 4，那么 1 号最少能拿 1，最多能拿 2；2 号最少能拿到 2，最多能拿到 3。

【输入输出样例 2】

tekken.in	tekken.out
1	-1
1	
1	
1 1 0 0	

【样例说明 2】

没有任何选手变更，第一届不可能收支不平衡。

【数据规模】

对于 10%的数据， $m=0$ ；

对于 20%的数据，薪金的范围形式为 $[x, x]$ ；

对于 50%的数据，满足 $1 \leq n \leq 5$ ， $m \leq 5$ ，薪金范围不超过 10；

对于 70%的数据，保证答案是整数。

对于 100%的数据， $n \leq 200$ ， $m \leq 100$ ，给定薪金范围不超过 20000。

OnePiece

(onepiece.pas/c/cpp)

【问题描述】

Monkey • D • Luffy 是立志要成为海贼王的男人（请自行断句），梦想就是找到 OnePiece。在大战役后的两年 Luffy 和他的伙伴们都通过打怪升级变强了许多，想要到新世界闯荡一番，找到 OnePiece。

Luffy 刚刚在鱼人岛干掉了全 OP 最二 Boss 霍迪，准备出发去找 OnePiece，这时一个路人 Kira 告诉了他一个小道消息，OnePiece 在新世界中距离鱼人岛最远的一个的岛那里。由于 Kira 非常诚实，Luffy 果断相信了，命令航海员 Nami 马上制定一个最快的去 OnePiece 的计划。

根据 Jinbe 的鲨鱼小弟带来的小道消息，新世界由 N 个岛组成，其中鱼人岛为 1 号岛，这些岛由 $N-1$ 条航路连接，任意两个岛之间都直接或间接连通。这 $N-1$ 条航路被分为 M 类，通过第 i 类航路只有两种可能的时间 A_i 和 B_i ，有 $P_i\%$ 的可能为 A_i ，若不为 A_i 则为 B_i 。对于一类从未通过的航路，只有通过一条该类的航路之后才知道通过这一类航路需要用的时间，通过同一类航路的时间相同。

制定计划时 Nami 问 Luffy 如果有多个最远的岛怎么办。众所周知，Luffy 是个白痴，他就说随便到一个就可以啦！

Nami 觉得制定这个计划太无聊了，于是去购物了，就把这个问题交给了另一个路人——你。Nami 的请求你不可能拒绝吧。Nami 给了你新世界的地图，要你在她购物回来后告诉她最优决策下到达 OnePiece 所用的期望时间。

【输入格式】

第一行两个正整数 N 和 M ，表示岛的个数和航路的种类数。

第二行 M 个正整数，第 i 个整数 A_i 表示通过第 i 种航路的第一种可能时间。

第三行 M 个正整数，第 i 个整数 B_i 表示通过第 i 种航路的第二种可能时间。

第四行 M 个正整数，第 i 个整数 P_i 表示通过第 i 种航路的为所用为 A_i 的概率 $P_i\%$ 。

接下来 $N-1$ 行，每行包含 3 个整数， U_i ， V_i ， C_i ，表示第 i 条边的两个端点为 U_i 和 V_i ，颜色为 C_i 。

【输出格式】

一行一个实数，表示最优决策下到达 OnePiece 所用的期望时间，误差不超过 $1e-6$ 。

【输入输出样例 1】

onepiece.in	onepiece.out
3 2 1 2 4 3 50 50 1 2 1 1 3 2	4.250000

【样例说明 1】

从 1 号岛出发,先去 2 号岛,知道通过第一类航路的时间,若为 4,则 OnePiece 在 2 号岛,若为 1,则 OnePiece 在 2 号岛。所用的期望时间为 $0.5*4+0.5*(1+1+0.5*3+0.5*2)=4.25$ 。

【输入输出样例 1】

onepiece.in	onepiece.out
10 3 3 10 4 1 2 1 50 50 50 2 1 2 3 1 2 4 2 1 5 2 1 6 5 1 7 4 2 8 1 3 9 8 2 10 5 1	16.125000

【数据规模】

对于 10%的数据, $M=1$;

对于 30%的数据, 满足 $1 \leq n \leq 10, m \leq 3$;

对于另外 20%的数据, 保证 $A_i=B_i$ 。

对于 100%的数据, $n \leq 50, m \leq 8, 0 \leq A_i, B_i \leq 1000, 1 \leq P_i \leq 99$ 。

【注意】

此题设有 spj。