2011 年全国青少年信息学奥林匹克
山东省省队选拔赛（第二轮）

第二试

竞赛时间：2011 年 4 月 17 日上午 8:00—12:00

题目名称	贪食蛇	保密	消耗战
目录	snake	tfnhnsm	repair
可执行文件名	snake.exe	tfnhnsm.exe	repair.exe
输入文件名	snake.in	tfnhnsm.in	repair.in
输出文件名	snake.out	tfnhnsm.out	repair.out
每个测试点时限	1s	1s	2s
测试点数目	20	10	10
每个测试点分值	5	10	10
内存限制	512MB	512MB	512MB
是否有部分分	无	无	无
题目类型	传统型	传统型	传统型

提交源程序须加后缀

对于 Pascal 语言	pas	pas	pas
对于 C　　语言	c	c	c
对于 C++　语言	cpp	cpp	cpp

注意：最终测试时，所有编译命令均不打开任何优化开关。

贪食蛇（snake）

【问题描述】

相信大家都玩过贪食蛇游戏，现在有一个改版贪食蛇游戏，跟传统的贪食蛇游戏一样，贪食蛇在活动区域内运动，吃食物，但是这个改版的贪食蛇游戏有着一些特别的规则。

活动区域：

贪食蛇的活动区域是一个 R 行 C 列的网格 A，贪食蛇活动不能超过这个网格的范围。第 i 行第 j 列的方格用 $A_{i,j}$ 表示。每个方格有一个整数权值，记作 $w(A_{ij})$。$0<=w(A_{ij})<=8$，$w(A_{ij})=0$ 时，A_{ij} 禁止进入；$w(A_{ij})>0$ 时，A_{ij} 允许进入。

方向：

对于 $P=(X_0,Y_0)$、$Q=(X_1,Y_1)$，有以下四种基本方向：

- 正左(L)：$X_0=X_1$ 且 $Y_0=Y_1-1$，则称 P 位于 Q 的正左方向。
- 正右(R)：$X_0=X_1$ 且 $Y_0=Y_1+1$，则称 P 位于 Q 的正右方向。
- 正上(U)：$X_0=X_1-1$ 且 $Y_0=Y_1$，则称 P 位于 Q 的正上方向。
- 正下(D)：$X_0=X_1+1$ 且 $Y_0=Y_1$，则称 P 位于 Q 的正下方向。

贪食蛇：

贪食蛇 B 是占据若干方格的图形，占据的方格数为贪食蛇的长度，记为 m，则贪食蛇从头到尾，用 B_1、B_2、……、B_m 表示。记 p 为贪食蛇的形态，若 Bi 位于第 X_i 行第 Y_i 列，则 $p(B_i)=(X_i,Y_i)$。初始情况下，m=4，且运动过程中始终需要满足以下限制：

- 对于 B_i 和 $B_{i+1}(1<=i<m)$，就是贪食蛇的前、后相邻两部分，必须满足 B_i 位于 B_{i+1} 的 L、R、U、D 四个方向之一。
- 对于 B_i 和 $B_j(1<=i<j<=m)$，$p(B_i)=(X_i,Y_i)$，$p(B_j)=(X_j,Y_j)$，需要满足 $X_i!=X_j$ 或 $Y_i!=Y_j$。也就是说，贪食蛇身体的任意一部分不能相交。

食物：

贪食蛇的活动区域内存在一些食物。每个食物位于一个允许进入的方格上，食物不会重叠。每个食物只能被吃一次。

贪食蛇的运动：

如果贪食蛇的头部 B_1 的 L、R、U、D 四个方向之一的 A_{ij} 能进入，且 A_{ij} 上不存在食物，则贪食蛇可以向该方向运动，新的头部位于 A_{ij} 上。记 p’为贪食蛇新的形态，则：

- $p'(B_k)=p(B_{k-1})$，当 $2<=k<=m$。
- $p'(B_k)=(i,j)$，当 k=1

贪食蛇的进食：

如果贪食蛇的头部 B_1 的 L、R、U、D 四个方向之一的 A_{ij} 能进入，且 A_{ij} 上存在食物，则贪食蛇可以向该方向进食，新的头部位于 A_{ij} 上，蛇的新长度

m’=m+1。记 p’为贪食蛇新的位置，则：

- $p'(B_k)=p(B_{k-1})$，当 2<=k<=m’。
- $p'(B_k)=(i,j)$，当 k=1

注意：运动或进食后的贪食蛇形态，仅仅需要考虑变换后的形态是否满足限制，不需要考虑变换的过程。也就是说，原来形态合法的贪食蛇的头部可以运动到尾部的位置，因为在变换后头部和尾部仍不会重叠。

运动或进食所需要的时间：

贪食蛇运动或进食，需要消耗时间。设运动或进食前头部所在的方格是 P，运动或进食后头部所在的方格是 Q，则此次运动或进食的所消耗的时间为 |w(P)-w(Q)|+1。

游戏的会在开始前给出贪食蛇的初始位置和所有食物的位置。你的任务是，以最少的时间令贪食蛇吃完所有食物。

【输入格式】

第一行，两个正整数 R、C。

接下来 R 行，每行 C 个没有空格分隔的数字。其中第 i 行第 j 个数字为 $w(A_{ij})$。

接下来 4 行，每行 2 个正整数。第 i 行的两个整数 X_i、Y_i，表示 $p(B_i)=(X_i,Y_i)$。

接下来一个正整数 N，表示食物的数量。

接下来 N 行，每行 2 个正整数 i、j，表示 A_{ij} 上存在一个食物。

【输出格式】

如果贪食蛇不能吃到所有的食物，输出“No solution.”（不包括引号）。

否则，输出：

第一行，一个整数，表示所需花费的时间；

第二行，一个由 L、R、U、D 组成的字符串，表示贪食蛇前进的方案。如果存在多种可能，你只需输出任意一种。

【样例输入】

```
5 5
11011
11011
11011
11011
11411
1 1
2 1
3 1
4 1
4
```

5 5
4 4
2 5
1 4

【样例输出】

21
RDDDDRRRULURULU

【数据规模和约定】

对于 20%的数据，N <= 1。
对于 40%的数据，N <= 2。
对于 60%的数据，N <= 3。
对于 100%的数据，N <= 4。

对于 30%的数据，R * C <= 36。
对于 100%的数据，R <= 12，C <= 12。

This filename has no special meanings

（tfnhnsm）

【问题描述】

现在，保密成为一个很重要也很困难的问题。如果没有做好，后果是严重的。比如，有个人没有自己去修电脑，又没有拆硬盘，后来的事大家都知道了。

当然，对保密最需求的当然是军方，其次才是像那个人。为了应付现在天上飞来飞去的卫星，军事基地一般都会建造在地下。

某 K 国的军事基地是这样子的：地面上两排大天井共 n1 个作为出入口，内部是许多除可以共享出入口外互不连通的空腔，每个空腔有且只有两个出入口，并且这两个出入口不会在同一排。为了方便起见，两排出入口分别编号为 1,3,5…和 2,4,6…并且最大的编号为 n1。

虽然上面扯了那么多关于保密的东西，但是其实解密也是一件很纠结的事情。但其实最简单直接暴力无脑的解密方法就是找个人去看看。。。

我们有很牛 X 的特种部队，只需要派出一支特种部队到 K 国基地的某个出入口，那么和这个出入口直接相连的所有空腔都可以被探索，但也只有这些空腔可以被这支部队探索。现在有足够多的特种部队可以供你调遣，你必须使用他们调查完所有的 K 国基地内的空腔。

当然，你的基地离 K 国基地不会太近，周边的地图将会给你，表示为 n 个检查点和 m 条连接这些点的道路，其中点 1 到点 n1 就是 K 国基地的出入口，点 n 是你的部队的出发点。对每条道路，有不同的通行时间 t 和安全系数 s。因为情报部门只对单向的道路安全系数进行了评估，所以这些道路只允许单向通行，并且不会存在环。

一支特种部队从你的基地出发，通过某条路径，到达某个 K 国基地出入口，此时这支部队的危险性表示为总时间和这条路径经过的所有道路的安全系数和的比值。整个行动的危险性表示为你派出的所有部队的危险性之和。你需要使这个值最小的情况下探索整个 K 国基地。

快点完成这个任务，在 K 国的叫兽宣布你是 K 国人之前。

【输入格式】

第一行 2 个正整数 n，m (4 <= n <= 700, m <= 100000) 表示整个地区地图上的检查点和道路数。

下面 m 行，每行 4 个正整数 a, b, t, s(a, b <=n, 1 <= t, s <= 10)表示一条从 a 到 b 的道路需时为 t，安全系数为 s。

接下来 1 行 2 个正整数 m1 和 n1(m1 <= 40000, n1 < min{n, 161}), m1 表示 K 国基地空腔的个数，n1 表示 K 国基地出入口的个数。

再接下来 m1 行，每行 2 个正整数 u, v (u, v<=n1, u 是奇数，v 是偶数)，表示每个空腔的 2 个出入口。

【输出格式】

一行，最小的危险性，保留一位小数。或者输出”-1”（无引号）表示此任务不可能完成。

【样例输入】

```
5 5
5 1 10 1
5 1 10 1
5 2 9 1
5 3 7 1
5 4 8 1
4 4
1 2
1 4
3 2
3 4
```

【样例输出】

```
17.0
```

【数据规模和约定】

对 30%的数据，n<=30
对 60%的数据，n<=300

另外 40%的数据 n1<=20

消耗战（repair）

【问题描述】

在一场战争中，战场由 n 个岛屿和 n-1 个桥梁组成，保证每两个岛屿间有且仅有一条路径可达。现在，我军已经侦查到敌军的总部在编号为 1 的岛屿，而且他们已经没有足够多的能源维系战斗，我军胜利在望。已知在其他 k 个岛屿上有丰富能源，为了防止敌军获取能源，我军的任务是炸毁一些桥梁，使得敌军不能到达任何能源丰富的岛屿。由于不同桥梁的材质和结构不同，所以炸毁不同的桥梁有不同的代价，我军希望在满足目标的同时使得总代价最小。

侦查部门还发现，敌军有一台神秘机器。即使我军切断所有能源之后，他们也可以用那台机器。机器产生的效果不仅仅会修复所有我军炸毁的桥梁，而且会重新随机资源分布（但可以保证的是，资源不会分布到 1 号岛屿上）。不过侦查部门还发现了这台机器只能够使用 m 次，所以我们只需要把每次任务完成即可。

【输入格式】

第一行一个整数 n，代表岛屿数量。
接下来 n-1 行，每行三个整数 u,v,w，代表 u 号岛屿和 v 号岛屿由一条代价为 c 的桥梁直接相连，保证 1<=u,v<=n 且 1<=c<=100000。
第 n+1 行，一个整数 m，代表敌方机器能使用的次数。
接下来 m 行，每行一个整数 k_i，代表第 i 次后，有 k_i 个岛屿资源丰富，接下来 k 个整数 $h_1,h_2,\ldots h_k$，表示资源丰富岛屿的编号。

【输出格式】

输出有 m 行，分别代表每次任务的最小代价。

【样例输入】

```
10
1 5 13
1 9 6
2 1 19
2 4 8
2 3 91
5 6 8
7 5 4
7 8 31
10 7 9
3
2 10 6
4 5 7 8 3
```

3 9 4 6

【样例输出】

12
32
22

【数据规模和约定】

对于 10%的数据，$2<=n<=10,1<=m<=5,1<=k_i<=n-1$
对于 20%的数据，$2<=n<=100,1<=m<=100,1<=k_i<=min(10,n-1)$
对于 40%的数据，$2<=n<=1000,m>=1,sigma(k_i)<=500000,1<=k_i<=min(15,n-1)$
对于 100%的数据，$2<=n<=250000,m>=1,sigma(k_i)<=500000,1<=k_i<=n-1$