

IOI2012 中国队选拔赛暨精英赛

CTSC 2012

第一试

竞赛时间：2012 年 5 月 7 日 8:00–13:00

题目名称	梭哈游戏	电阻网络	最短路
目录	showhand	circuit	shortest
可执行文件名	showhand	circuit	N/A
输入文件名	showhand.in	circuit.in	shortest1.in~ shortest10.in
输出文件名	showhand.out	circuit.out	shortest1.out~ shortest10.out
每个测试点时限	5s	4s	N/A
内存限制	256MB	256MB	N/A
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	是
题目类型	传统型	传统型	提交答案
附加文件	无	无	有

提交源程序须加后缀

对于 C++ 语言	showhand.cpp	circuit.cpp	N/A
对于 C 语言	showhand.c	circuit.c	N/A
对于 Pascal 语言	showhand.pas	circuit.pas	N/A

注意：最终测试时，所有编译命令均不打开任何优化开关。

梭哈游戏

【问题描述】

小 Y 和小 Z 最近迷上了一种叫梭哈的扑克游戏。梭哈又称沙蟹，是英文 Show Hand 的音译，是一种使用黑桃、红心、梅花、方片的 A 到 K 共 52 张牌(没有大小王)来进行的扑克牌游戏。

和其他扑克游戏一样，梭哈的目的是得到最大的牌型并赢得牌局：每名玩家首先需要下基本的注额，之后将获得一张底牌，这张牌只有自己知道。普通梭哈游戏的规则是，在发完底牌后的第一轮时，每个玩家都将得到一张明牌（明牌是摆在台面上的，所有人都能看见），拥有最大明牌的玩家首先发言，他可以下注、不下注（让牌）或盖牌（放弃）也可以全压（梭哈），其他玩家可以跟注（有玩家全压时必须全压）、加注或盖牌（放弃），放弃的玩家将无法继续游戏，并且之前押的筹码无法取回。而全压之后将直接把每个人的手牌补充至 5 张进行最后的判定；第二圈、第三圈和第四圈的进程与第一圈是类似的。最后，每位玩家要比牌型的大小以确定赢家。牌最大的玩家赢得牌局。

所有五张牌的组合，按以下秩序，由大至小排行分为不同牌型：

- 1、同花顺 (Straight Flush)：同一花色，顺序的牌。例：Q♠ J♠ 10♠ 9♠ 8♠；
- 2、四条 (Four of a Kind)：有四张同一点数的牌。例：10♣ 10♦ 10♥ 10♠ 9♥；
- 3、满堂红 (Full House)：三张同一点数的牌，加一对其他点数的牌。例：8♣ 8♦ 8♠ K♥ K♠；
- 4、同花 (Flush)：五张同一花色的牌。例：A♠ K♠ 10♠ 9♠ 8♠；
- 5、顺子 (Straight)：五张顺连的牌。例：K♦ Q♥ J♠ 10♦ 9♦；
- 6、三条 (Three of a kind)：有三张同一点数的牌。例：J♣ J♥ J♠ K♦ 9♠；
- 7、两对 (Two Pairs)：两张相同点数的牌，加另外两张相同点数的牌。例：A♣ A♦ 8♥ 8♠ Q♠；
- 8、一对 (One Pair)：两张相同点数的牌。例：9♥ 9♠ A♣ J♠ 8♥；
- 9、无对 (Zilch)：不能排成以上组合的牌，以点数决定大小。例：A♦ Q♦ J♠ 9♣ 8♣。

若牌型一样则 利用点数和花色决定胜负。（点数优先）

点数的顺序 (从大至小) 为：A>K>Q>J>10>9>8>7>6>5>4>3>2。（注：当 5 张手牌是 5 4 3 2 A 的时候，A 可以看作最小的牌，此时的牌型仍然为顺子，是顺子里面最小的一个）。

花色的顺序 (大至小) 为：黑桃(♠)>红心(♥)>梅花(♣)>方块(♦)。

举例说明：

- 1、Q♦ J♦ 10♦ 9♦ 8♦ > 8♣ 8♥ 8♠ K♥ K♠ （前者牌型为同花顺，比后者大）；
- 2、9♣ 9♦ 9♠ Q♥ Q♠ > 8♣ 8♦ 8♠ K♥ K♠ （两者牌型均为满堂红，比较牌型中三张同一点数的牌 9 比 8 大）；
- 3、A♣ A♦ 8♥ 8♠ Q♠ > A♠ A♥ 7♥ 7♠ K♠ （两者牌型均为两对，且最大的对子相同，此时比较次大的对子，8 比 7 大）；
- 4、A♠ Q♠ J♥ 9♥ 8♥ > A♦ Q♦ J♠ 9♣ 8♣ （两者牌型均为无对，所有数码均相同，此时比较最大牌的花色，A♠ > A♦）。
- 5、4♣ 4♥ A♦ Q♦ 5♦ > 4♣ 4♦ A♠ Q♠ 5♠ （两者牌型均为一对，所有数码均相

同，此时对 4 为牌型里最大的部分，因此比较 $4\spadesuit > 4\clubsuit$ ）

在小 Y 和小 Z 玩梭哈的过程中，小 Y 总希望能够实时了解目前的形势，即根据自己手上的牌算出自己的胜率。但是他的编程能力有限，你能帮他完成这个任务么？

【输入格式】

输入文件为 showhand.in。

第一行包含 1 个正整数 N ，表示小 Y 自己手上牌的数量。

接下来 N 行每行用两个整数描述一张小 Y 手上的牌：第一个数表示牌的数码（1 表示 A，13 表示 K，12 表示 Q，11 表示 J），第二个数表示牌的花色（1 表示黑桃，2 表示红心，3 表示梅花，4 表示方块）。

接下来 $N-1$ 行每行用两个整数描述一张小 Z 手上的明牌：第一个数表示牌的数码（1 表示 A，13 表示 K，12 表示 Q，11 表示 J），第二个数表示牌的花色（1 表示黑桃，2 表示红心，3 表示梅花，4 表示方块）。

【输出格式】

输出文件 showhand.out 仅包含一行，格式为 A/B 形式。A,B 为两个互质的自然数，A/B 表示了小 Y 当前局面下的获胜可能性。特别的，若 A 为 0，则应输出 0/1。

【样例输入】

```
5
2 1
2 2
2 3
2 4
3 1
1 1
1 2
1 3
3 2
```

【样例输出】

```
42/43
```

【样例说明】

小 Y 手上牌的牌型是 4 条，小 Z 若底牌是方块 A 则也能构成 4 条，并通过数码大获胜。其他情况下都是小 Y 获胜。

【数据规模和约定】

对于 10% 的数据， $N = 5$ ；

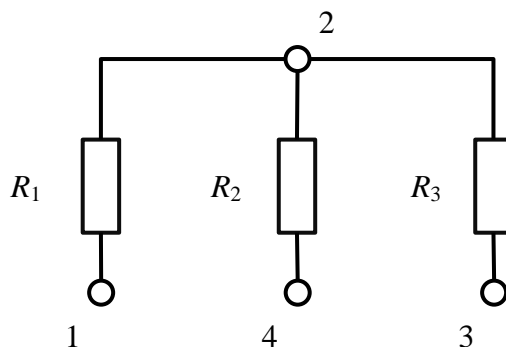
对于 30% 的数据， $3 \leq N \leq 5$ ；

对于 100% 的数据， $1 \leq N \leq 5$ 。

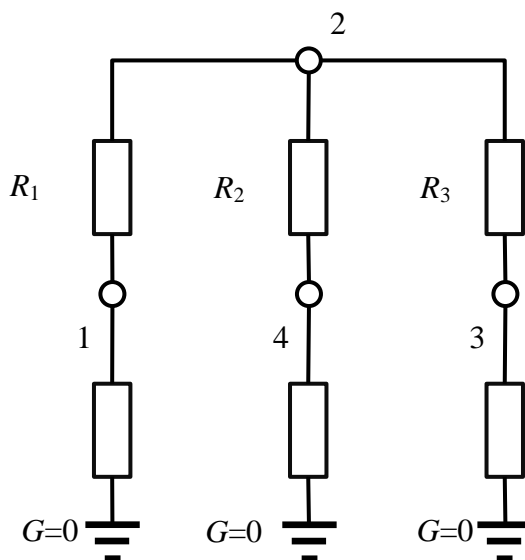
电阻网络

【问题描述】

给定一个树型电网，树中的每条边上有一个电阻 R_i ，电阻值均为 10000Ω 。下图为一个包含 4 个节点的树型电路情况：

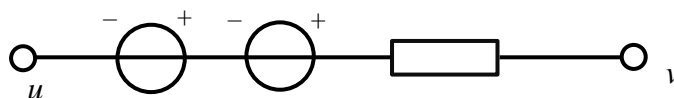


树中的所有叶子节点（度为 1 的节点称为叶子节点）都接地，每条接地线上都附有 10000Ω 的电阻，最终形成的电网如下图所示：



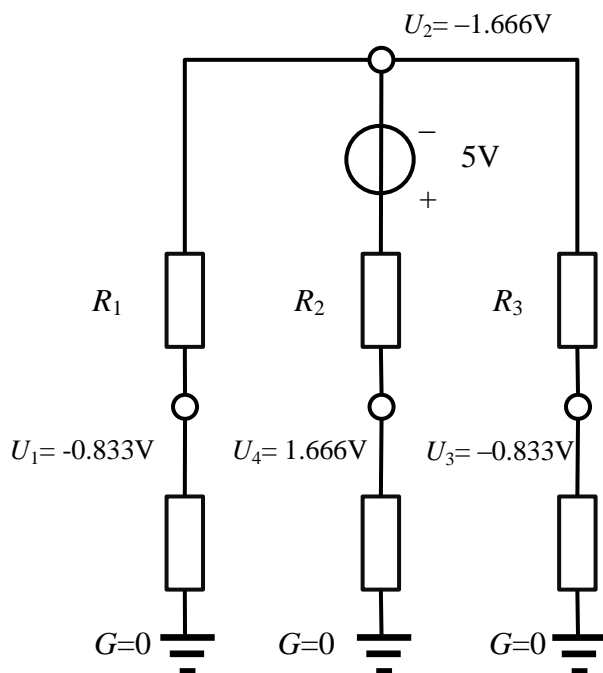
现有如下两种操作：

$C u v w$ ：表示在边 $\langle u, v \rangle$ 上串联一个电源，电源的大小为 w 伏，电源位于靠近节点 u 一侧（如下图所示），电源负极指向 u 。注意同一条边上可以串联多个电源。



$Q u$: 表示询问点 u 当前的电压, 此电压是指对地电压。

如对上述进行 C 2 4 5 操作后, 网络变为:



此时每个节点上的电压见上图的标注。

【输入格式】

输入文件 `circuit.in` 的第一行包含两个整数 N, M , 分别表示树的节点数和操作个数。接下来 $N-1$ 行, 每行两个数 u, v , 表示有一条连接节点 u, v 的边, 这条边上恰好包含一个电阻。

接下来 M 行, 每行一个命令, 格式见题目描述。

【输出格式】

输出文件为 `circuit.out`。对于每个 Q 命令, 输出一个数表示此刻该点的电压值。你可以输出任意多位的小数, 只要你的答案和标准答案相差不超过 10^{-3} 就算合法。

【样例输入】

```

4 3
1 2
2 3
2 4
Q 2
C 2 4 5
Q 2

```

【样例输出】

```

0.0000000000
-1.6666666666

```

【样例说明】

对于第一个询问，由于原图中没有电源，所以没有电流，所有点的电压都相等（否则如果有 $U_i > U_j$ ，则就有 i 流向 j 的电流，与没有电源矛盾），都等于地电压 $0V$ 。

之后在 $\langle 2,4 \rangle$ 中加一个 $5V$ 的电源，得到的新图见题目描述。

整理后可以发现，新图的形式是串联(电源, $R_2+10000$), 并联($R_1+10000$, $R_3+10000$)), 由此可以得到新图的总电阻为:

$$R_2+10000+1/(1/(R_3+10000)+1/(R_1+10000))=30000 \Omega.$$

所以流过节点 4 的电流就是 $5/30000A$ ，所以 $U_4=5/3V$ 。 $U_2=U_4+R_2*I=5/3V$ ，由于 U_1 和 U_3 形式对称，由分压关系可知 $U_1=U_3=U_2*10000/(10000+10000)=-5/6V$ 。

【数据规模】

30%的数据保证 $N, M \leq 30$

60%的数据保证 $N, M \leq 3000$

100%的数据保证 $3 \leq N, M \leq 50000$, $1 \leq u, v \leq n$, $1 \leq w \leq 10$, 树中最长链的长度不超过 50。

最短路

【问题描述】

给定一个节点 1 和节点 N 连通的正权无向图 G ，请你删除 不超过 K 条边，使得节点 1 和节点 N 仍然连通的同时，且这两点之间的 最短路尽可能长。

【输入格式】

本题为提交答案试题，输入文件 `shortest1.in~shortest10.in` 已经在选手目录下。

输入文件 `shortest*.in` 的第一行包含三个正整数 N ， M 和 K 。其中 N 表示节点数， M 表示边数，节点的编号由 1 至 N ，边的编号由 1 至 M 。接下来 M 行，每行三个正整数 u ， v 和 w ，表示有一条连接节点 u 和节点 v 的边，权值为 w 。

【输出格式】

输出文件 `shortest*.out` 的第一行包含一个非负整数 T ($T \leq K$)，表示需要删掉的边数。

接下来 T 行，每行一个 1 到 M 之间的整数 x ，表示删掉输入中的第 x 条边。你需要保证这 T 个整数互不相同。

【样例输入】

```
3 3 1
1 2 1
2 3 1
1 3 1
```

【样例输出】

```
1
3
```

【样例说明】

样例中从节点 1 到 3 的最短路径长度为 1，删去第三条边之后，最短路径长度为 2。

【评分标准】

对于每个测试点，设有评分四个参数 s_1, s_2, s_3, s_4 。假设你的方案的最短路为 ans 。

如果你没有输出，或者输出不合法，或者最短路不存在，得 0 分。

如果最短路存在，得 1 分。

如果 $ans \geq s_1$ ，得 3 分。

如果 $ans \geq s_2$, 得 5 分。

如果 $ans \geq s_3$, 得 8 分。

如果 $ans = s_4$, 得 10 分。

如果 $ans > s_4$, 得 12 分。

取满足条件的分数中的最高得分为该测试点你的得分。

【如何测试你的输出】

在你的目录下有一个名为 `checker` 的程序可以用来检查你的输出，你可以在终端中使用以下命令来检查你的输出：

```
./checker N
```

其中 N 为测试点的编号，例如，要测试第 3 个测试点可以使用

```
./checker 3
```

该程序会检测你的输出方案是否合法。如果方案合法，程序还会给出该方案的最短路的长度值。

【特别提示】

请妥善保存输入文件 `*.in` 和你的输出 `*.out`，及时备份，以免误删。