

CCF 全国信息学奥林匹克联赛（NOIP2018）复赛

提高组 Day2 模拟赛强化训练

（请选手务必仔细阅读本页内容）

一. 题目概况

中文题目名称	麻将	挤奶	瓦解
英文题目与子目录名 (Lemon 测评机请忽略)	mahjong	milkorder	disruption
可执行文件名	mahjong	milkorder	disruption
输入文件名	mahjong.in	milkorder.in	disruption.in
输出文件名	mahjong.out	milkorder.out	disruption.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	10	8	10
附加样例文件	见附件	见附件	见附件
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统
运行内存上限	256M	256M	256M

二. 交源程序文件名

对于 C++语言	mahjong.cpp	milkorder.cpp	disruption.cpp
对于 C 语言	mahjong.c	milkorder.c	disruption.c
对于 pascal 语言	mahjong.pas	milkorder.pas	disruption.pas

三. 编译命令（不包含任何优化开关）

对于 C++语言	g++ -o mahjong mahjong.cpp -lm	g++ -o milkordermilkorder.cpp -lm	g++ -o disruptiondisruption.cpp -lm
对于 C 语言	gcc -o mahjong mahjong.c -lm	gcc -o milkordermilkorder.c -lm	gcc -o disruptiondisruption.c -lm
对于 pascal 语言	fpc mahjong.pas	fpc milkorder.pas	fpc disruption.pas

注意事项:

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 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、测评环境为 Window10
- 4、特别提醒：评测在 Lemon 下进行，各语言的编译器版本以其为准。

1. 麻将

(mahjong .cpp/c/pas)

【问题描述】

麻将是中国传统的娱乐工具之一。麻将牌的牌可以分为字牌（共有东、南、西、北、中、发、白七种）和序数牌（分为条子、饼子、万子三种花色，每种花色各有一到九的九种牌），每种牌各四张。在麻将中，通常情况下一组和了的牌（即完成的牌）由十四张牌组成。十四张牌中的两张组成对子（即完全相同的两张牌），剩余的十二张组成三张一组的四组，每一组须为顺子（即同花色且序数相连的序数牌，例如条子的三、四、五）或者是刻子（即完全相同的三张牌）。一组听牌的牌是指一组十三张牌，且再加上某一张牌就可以组成和牌。那一张加上的牌可以称为等待牌。在这里，我们考虑一种特殊的麻将。在这种特殊的麻将里，没有字牌，花色也只有一种。但是，序数不被限制在一到九的范围内，而是在 1 到 n 的范围内。同时，也没有每一种牌四张的限制。一组和了的牌由 $3m + 2$ 张牌组成，其中两张组成对子，其余 $3m$ 张组成三张一组的 m 组，每组须为顺子或刻子。现给出一组 $3m + 1$ 张的牌，要求判断该组牌是否为听牌（即还差一张就可以和牌）。如果是的话，输出所有可能的等待牌。

【输入】

包含两行。第一行包含两个由空格隔开整数 n, m ($9 \leq n \leq 400, 4 \leq m \leq 1000$)。第二行包含 $3m + 1$ 个由空格隔开整数，每个数均在范围 1 到 n 之内。这些数代表要求判断听牌的牌的序数。

【输出】

输出为一行。如果该组牌为听牌，则输出所有的可能的等待牌的序数，数字之间用一个空格隔开。所有的序数必须按从小到大的顺序输出。如果该组牌不是听牌，则输出“NO”。

【输入输出样例 1】

mahjong .in	mahjong .out
9 4 1 1 2 2 3 3 5 5 5 7 8 8 8	6 7 9

【数据范围】

对于 100%的数据， $9 \leq n \leq 400, 4 \leq m \leq 1000$

2. 挤奶

(milkorder.cpp/c/pas)

【问题描述】

Farmer John 的 N 头奶牛 ($1 \leq N \leq 10^5$)，仍然编号为 $1 \cdots N$ ，正好闲得发慌。因此，她们发展了一个与 Farmer John 每天早上为她们挤牛奶的时候的排队顺序相关的复杂的社会阶层。经过若干周的研究，Farmer John 对他的奶牛的社会结构总计进行了 M 次观察 ($1 \leq M \leq 50,000$)。每个观察结果都是他的某些奶牛的一个有序序列，表示这些奶牛应该以与她们在序列中出现的顺序相同的顺序进行挤奶。比方说，如果 Farmer John 的一次观察结果是序列 2、5、1，Farmer John 应该在给奶牛 5 挤奶之前的某个时刻给奶牛 2 挤奶，在给奶牛 1 挤奶之前的某个时刻给奶牛 5 挤奶。Farmer John 的观察结果是按优先级排列的，所以他的目标是最大化 X 的值，使得他的挤奶顺序能够符合前 X 个观察结果描述的状态。当多种挤奶顺序都能符合前 X 个状态时，Farmer John 相信一个长期以来的传统——编号较小的奶牛的地位高于编号较大的奶牛，所以他会最先给编号最小的奶牛挤奶。更加正式地说，如果有多个挤奶顺序符合这些状态，Farmer John 会采用字典序最小的那一个。挤奶顺序 x 的字典序比挤奶顺序 y 要小，如果对于某个 j ， $x_i = y_i$ 对所有 $i < j$ 成立，并且 $x_j < y_j$ （也就是说，这两个挤奶顺序到某个位置之前都是完全相同的，在这个位置上 x 比 y 要小）。请帮助 Farmer John 求出为奶牛挤奶的最佳顺序。

【输入】

第一行包含 N 和 M 。接下来的 M 行，每行描述了一个观察结果。第 $i+1$ 行描述了观察结果 i ，第一个数是观察结果中的奶牛数量 m_i ，后面是一列 m_i 个整数，给出这次观察中奶牛的顺序。所有 m_i 的和至多为 200,000

【输出】

输出 N 个空格分隔的整数，给出一个 $1 \cdots N$ 的排列，为 Farmer John 给他的奶牛们挤奶应该采用的的顺序。

【输入输出样例】

milkorder.in	milkorder.out
4 3 3 1 2 3 2 4 2 3 3 4 1	1 4 2 3

【数据范围】

对于 30% 的数据， $1 \leq N \leq 10^3$ ， $1 \leq M \leq 50000$

对于 30% 的数据， $1 \leq N \leq 10^5$ ， $1 \leq M \leq 500$

对于 100% 的数据， $1 \leq N \leq 10^5$ ， $1 \leq M \leq 50000$

【样例说明】

这里，Farmer John 有四头奶牛，他的挤奶顺序应该是奶牛 1 在奶牛 2 之前、奶牛 2 在奶牛 3 之前（第一个观察结果），奶牛 4 在奶牛 2 之前（第二个观察结果），奶牛 3 在奶牛 4 之前、奶牛 4 在奶牛 1 之前（第三个观察结果）。前两个观察结果可以同时被满足，但是 Farmer John 不能同时满足所有的规则，因为这样的话会要求奶牛 1 在奶牛 3 之前，同时奶牛 3 在奶牛 1 之前。这意味着总共有两种可能的挤奶顺序：1 4 2 3 和 4 1 2 3，第一种是字典序较小的。

3. 瓦解

(disruption.cpp/c/pas)

【问题描述】

Farmer John 自豪于他所经营的交通发达的农场。这个农场是由 N 块牧场 ($2 \leq N \leq 50,000$) 组成的， $N-1$ 条双向道路将它们连接起来，每一条道路的都为一单位长度。Farmer John 注意到，从任何一块牧场到另一块牧场，都能通过一组合适的道路到达。

尽管 FJ 的农场现在是连通的，他担心如果有一条道路被阻断会发生什么，因为这事实上会将他的农场分为两个不相交的牧场集合，奶牛们只能够在每一个集合内移动但不能在集合间移动。于是 FJ 又建造了 M 条额外的双向道路 ($1 \leq M \leq 50,000$)，每一条的长度都是一个至多为 10^9 的正整数。奶牛们仍然可以使用原有的道路进行移动，除非其中的某些被阻断了。

如果某条原有的道路被阻断了，农场就会被分为两块不相交的区域，那么 FJ 就会从他的额外修建的道路中选择一条能够重建这两块区域的连通性的，取代原来那条，从而奶牛们又可以从任何一块牧场去往另一块牧场。对于农场上每一条原有的道路，帮助 FJ 选出最短的替代用的道路。

【输入】

输入的第一行包含 N 和 M 。接下来的 $N-1$ 行，每行用整数 p 和 q 描述了一条原有的道路，其中 $p \neq q$ 是这条道路连接的两块牧场（在 $1 \cdots N$ 范围内）。剩下的 M 行，每行用三个整数 p 、 q 和 r 描述了一条额外的道路，其中 r 是这条道路的长度。任何两块牧场之间至多只有一条道路

【输出】

对原有的 $N-1$ 条道路的每一条，按照它们在输入中出现的顺序，输出如果这条道路被阻断的话，能够重新连接农场的最短的替代用道路的长度。如果不存在合适的替代用的道路，输出 -1 。

【输入输出样例 1】

disruption.in	disruption.out
6 3	7
1 2	7
1 3	8
4 1	5
4 5	5
6 5	
2 3 7	
3 6 8	
6 4 5	

【数据范围】

对于 30% 的数据， $2 \leq N \leq 500$ $1 \leq M \leq 5000$

对于 100% 的数据， $2 \leq N \leq 50000$ $1 \leq M \leq 50000$