

# 简化树的同构

(isomorphism.pas/c/cpp)

## 【问题描述】

一个无向树的度数为 2 的结点称为假结点，其它结点称为真结点。一个无向树的简化树其结点由原树的全体真结点组成，两个真结点之间有边当且仅当它们在原树中有边，或者在原树中有一条联结这两个结点的路，其中间节点全是假结点。两个无向树各自的简化树如果同构，即存在结点之间的一一对应，使得在一个树中的任意两个结点之间有边当且仅当它们的对应结点在另一个树中有边，则称原来的两个无向树实质同构。给定若干个无向树，将相互实质同构的无向树只保留一个其余删除。统计剩下的相互不实质同构的无向树个数，并将它们的简化树结点个数从小到大输出。

### 【输入格式】

第一行只有一个正整数  $m$ , 后面依次输入  $m$  个无向树, 每个无向树先用一行输入结点个数  $n$ , 结点就用 1 到  $n$  表示, 然后用  $n-1$  行输入  $n-1$  条无向边, 每行有两个 1 到  $n$  之间的不同的正整数, 用一个空格隔开, 代表这两个结点之间有无向边。两个树之间无空行。

### 【输出格式】

第一行输出一个正整数, 即输入中不计实质同构包含无向树的个数  $m_0$  ( $1 \leq m_0 \leq m$ )。第二行包含不严格递增的  $m_0$  个正整数, 表示这  $m_0$  个无向树的简化树结点个数。相邻两数用一个空格隔开。

### 【样例输入】

isomorphism.in

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

### 【样例输出】

isomorphism.out

```
1
4
```

### 【数据规模】

30%的数据满足:  $2 \leq m \leq 10$ ,  $2 \leq n \leq 10$

60%的数据满足:  $2 \leq m \leq 20$ ,  $2 \leq n \leq 300$

100%的数据满足:  $2 \leq m \leq 20$ ,  $2 \leq n \leq 10000$