

**Notice:** 1:由于本OJ建立在Linux平台下，而许多题的数据在Windows下制作，请注意输入、输出语句及数据类型及范围，避免无谓的RE出现。 2:本站即将推出针对初学者的试题系统(与目前OJ是分开的，互不影响)，内容覆盖从语法入门到NOI的所有知识点，敬请关注。

## 1194: [HNOI2006]潘多拉的盒子

Time Limit: 10 Sec Memory Limit: 162 MB

Submit: 295 Solved: 160

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

### Description

传说中，有个神奇的潘多拉宝盒。如果谁能打开，便可以拥有幸福、财富、爱情。可是直到真的打开，才发现与之相随的还有灾难、不幸。

其实，在潘多拉制造这个宝盒的时候，设置了一些咒语来封锁住灾难与不幸。然而，直到科技高度发达的今天，人们才有希望弄懂这些咒语。所以说，上千年来，人们只得忍受着各种各样的疾病和死亡的痛苦。

然而，人类的命运从此改变了。经过数十年的研究，NOI 组织在最近终于弄清楚了潘多拉咒语的原理。

咒语是由一个叫做咒语机的机器产生的。用现在的名词来解释，咒语机其实就是一个二进制产生器，它产生的一个二进制字符串（这个字符串叫做咒语源）经加密后就形成了咒语。二进制产生器的结构是这样的：

它由  $n$  个元件组成，不妨设这  $n$  个元件的标号为  $0$  到  $n-1$ 。在每个时刻，都有且仅有一个信号，它停留在某个元件上。一个信号就是一个二进制字符串。最开始，有一个空串信号停留在元件  $0$  上。在某个时刻，如果有一个信号  $s$  停留在元件  $i$  上，那么，这时元件  $i$  可以将信号后面加一个  $0$ ，然后把信号传给元件  $p_i, 0$ ，也可以将信号后面加一个  $1$ ，然后传给元件  $p_i, 1$ 。也就是说，下一个时刻有可能，一种可能是一个信号  $S0$ （表示字串  $S$  后面加一个  $0$  形成的字串）仅在元件  $p_i, 0$  上，另一种可能是有一个信号  $S1$  停留在元件  $p_i, 1$  上。

有的元件可以将停留在它上面的信号输出，而输出的信号就成为了咒语源，这样的元件叫做咒语源输出元。

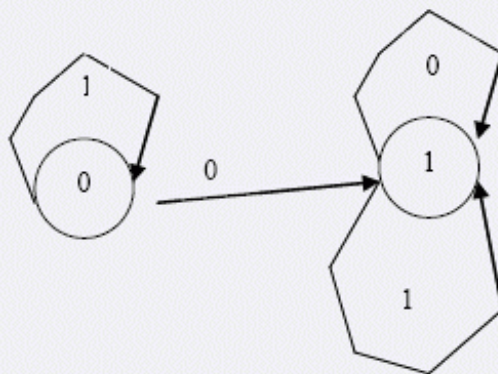
不难发现，有些咒语源是可能由一个咒语机产生的，而另一些咒语源则不行。

例如，右图的咒语机能产生 1, 11, 111, 1111, ……等咒语源，但是不能产生 0, 10, 101 等咒语源。

在这个盒子上，有  $K$  个咒语机，不妨将这些咒语机从 0

到  $K-1$  标号。可能有这种情况，一个咒语机  $i$  能够产生的咒语源，咒语机  $j$  都能产生。这时，我们称咒语机  $j$  是咒语机  $i$  的升级。而衡量这个例子的复杂程度的一种办法是：看这个盒子上升级次数最多的一个咒语机。即：找到一个最长的升级序列  $a_1, a_2, \dots, a_t$ 。该升级序列满足：序列中任意两个咒语机的标号都不同，且都是 0 到  $k-1$ （包含 0 和  $k-1$ ）之间的整数，且咒语机  $a_2$  是咒语机  $a_1$  的升级，咒语机  $a_3$  是咒语机  $a_2$  的升级，……，咒语机  $a_t$  是咒语机  $a_{t-1}$  的升级。

你想远离灾难与不幸吗？你想从今以后沐浴幸福的阳光吗？请打开你的潘多拉之盒吧。不过在拱形它之前，你先得计算一下宝盒上最长的升级序列。



图中,0 是咒语源输出元.箭头指示了  $p$ 。例如图中,  
 $p_{0,1}=0$ ,  $p_{0,0}=1$ ,  
 $p_{1,0}=1$ ,  $p_{1,1}=1$ 。

## Input

第一行是一个正整数  $S$ ，表示宝盒上咒语机的个数， $(1 \leq S \leq 50)$ 。文件以下分为  $S$  块，每一块描述一个咒语机，按照咒语机 0, 咒语机 1, ..., 咒语机  $S-1$  的顺序描述。每一块的格式如下。一块的第一行有两个正整数  $n, m$ 。分别表示该咒语机中元件的个数、咒语源输出元的个数  $(1 \leq m \leq n \leq 50)$ 。接下来一行有  $m$  个数，表示  $m$  个咒语源输出元的标号（都在 0 到  $n-1$  之间）。接下来有  $n$  行，每一行两个数。第  $i$  行  $(0 \leq i \leq n-1)$  的两个数表示  $p_{i,0}$  和  $p_{i,1}$ （当然，都在 0 到  $n-1$  之间）。

## Output

第一行有一个正整数  $t$ , 表示最长升级序列的长度。

## Sample Input

```
4
1 1
0
0 0
2 1
```

0

1 1

0 0

3 1

0

1 1

2 2

0 0

4 1

0

1 1

2 2

3 3

0 0

## Sample Output

3

## HINT

## Source

[\[Submit\]](#)[\[Status\]](#)[\[Discuss\]](#)

[HOME](#) [Back](#)

[한국어](#) [中文](#) [فارسی](#) [English](#) [ไทย](#)

版权所有 ©2008-2012 大视野在线测评 | 湘ICP备13009380号 | 站长统计

Based on opensource project hustoj.