# 无向图tarjan求割边割点、最近公共祖先总结

**割点：删除这个点之后整个图变成不连通的两个部分的点**

**割点集合：在一个无向图中删除该集合中的所有点，能使原图变成互不相连的连通块的点的集合**

**点连通度：最小割点集合点数**

**割边（桥）：类似于割点，删除一条边后会使一个连通图变得不完全连通**

**割边集合：所有割边的集合**

**边连通度：最小割边集合边数**

**与有割边割点的图对应，有一种无向图，它的边连通度（或点连通度）大于1，那么我们称它为双连通分量，即删掉任意点（或边），图仍然是连通的**

**用tarjan算法求割点**

**原理：在一棵DFS树中**

**根root是割顶当且仅当它至少有两个儿子**

**其他点v是割顶当且仅当它有一个儿子u, 从u或者u的后代出发没有指向v祖先(不含v)的B边, 则删除v以后u和v的父亲不连通, 故为割顶**

1. **基本算法同tarjan经典算法**
2. **每遍历一个新的（颜色为白色）u的儿子v都记录个数**
3. **low[u]值更新后进行以下判断（前提v未被遍历过）：**
4. **u为树根，且儿子个数大于1**
5. **u不为树根，但low[v]>=dfn[u]（说明v及其子节点都不能达到u以上的父亲节点）**

**满足以上任意条件u便为割点，记录在数组里，tarjan完成后再输出（中途输出会重复）**

**用tarjan算法求割边（桥）**

**原理**

**发现T边(u,v)时若发现v和它的后代不存在一条连接u或其祖先的B边, 则删除(u,v)后u和v不连通, 因此(u,v)为桥**

**桥的判定算法**

**发现T边(u, v)时若low[v]>=dfn[u],则(u,v)为桥**

1. **基本算法同tarjan经典算法**
2. **形参加上father，作用为记录u的父亲节点，避免在遍历v时遇到重边重新更新low[u]**
3. **在“v已被遍历”的位置加上判断 如果v点等于father，那么就不执行更新low[u]值**
4. **Tarjan算法结束后，遍历所有节点u及其子节点v，如果low[u]==low[v]，那么这条边就为割边**

**求点双连通分支(去掉割点之后所得的双连通分量)**

1. **tarjan求割点**
2. **每找到一个割点，将它上面的所有点弹出栈，所得的点集就是双连通分量**

**求边双连通分支**

1. **tarjan找桥边**
2. **删除桥边（1、2两步同时完成）**
3. **剩余各部分则为双连通分量**

**有桥的连通图，加边变成双连通图**

1. **tarjan找桥边并删除桥边**
2. **将点双连通分量收缩成为一个顶点**
3. **加回桥边，统计度为1的顶点个数**
4. **根据规律，需要加入的桥边最少为(n+1)/2条**

**最近公共祖先LCA**

**原理：在一棵树中，父亲结点与儿子结点的最近公共祖先为父亲结点**

1. **建立并查集**
2. **DFS先序遍历一棵树**
3. **在每个结点上，进行以下操作：**
4. **涂黑该结点**
5. **如果有关于该结点a的询问，搜索另一询问的结点b，如果b也被涂黑，那么它们的公共祖先为b所在并查集的祖先，输出结果**

**拓展：求一棵树中两个结点的距离**

1. **建立dis[]数组，记录从树根到i点的距离dis[i]**
2. **则a与b的距离为**

**d=dis[a]-dis[LCA]+dis[b]-dis[LCA]=dis[a]+dis[b]-2\*dis[LCA]**

**Poj 1144**

**描述**

一个电话线公司（简称TLC）正在建立一个新的电话线缆网络。他们连接了若干个地点分别从1到N编号。没有两个地点有相同的号码。这些线是双向的并且能使两个地点保持通讯。每个地点的线都终结于电话交换机。每个地点都有一个电话交换机。从每个地点都能通过线缆到达其他任意的地点，然而它并不需要直接连接，它可以通过若干个交换机来到达目的地。有时候某个地点供电出问题时，交换机就会停止工作。TLC的工作人员意识到，除非这个地点是不可达的，否则这种情况就会发生，它还会导致一些其它的地点不能互相通讯。在这种情况下我们会称这个地点（错误发生的地方）为critical。现在工作人员想要写一个程序找到所有critical地点的数量。帮帮他们。

**输入**

输入文件包括若组测试数据。每一组是一个网络，每一组测试数据的第一行是地点的总数量N<100. 每个接下来最多N行包括一个数字表示一个地点和与它相连接的地点的数字。这些最多N行完全描述了整个网络，比如，网络中每个直接连接的两个地点被至少一行包括。一行内的所有数字都要用空格隔开。每组数据需要用单独的一个0结束。最后的块只有一行即N=0。

**输出**

输出除了最后一个组其他每一个组的critical地点的数量，每个块用一行输出。

**样例输入：**

5

5 1 2 3 4

0

6

2 1 3

5 4 6 2

0

0

**样例输出**

1

2

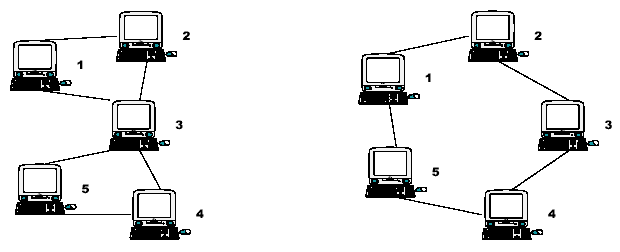
**提示：**

你需要确定每行的结束。为了方便判断，每行的结束都没有多余的空白

**读入是难点，其余就是找割点的部分**

**POJ1523**

**描述**  
考虑一下下图所示的两个网络。假设在这些网络中数据只能在两个直接连接的节点中传输。在左图网络中一个单独节点3的错误就会阻止一些其它可用节点的互相通信。节点1和2可以保持通信，4和5也是如此，但是其他的配对就无法再保持通信了。



因此3节点就是这个网络中的Single Point of Failure（SPF）。严格来讲，如果一个任意的不可用节点会阻止原有全通网络中至少一对可用节点保持通信，那么这就是一个SPF。注意在右图的网络中并没有这种节点。因此这个网络就没有SPF。至少有两台机器需要错误才能导致这个网络中有可用的节点对无法通信。

**输入**

输入会包括若干网络的描述。一个网络描述会包括若干对的整数，每行一对整数来描述节点的连接情况，先后顺序是无关的，如：1 2和2 1描述了相同的连接。所有的节点编码会从1到1000.一个单独的0行来结束连接节点的列表。一个空的网络描述结束输入。输入文档中空白的行要被忽略。

**输出**

对于每个输入的网络中，你需要输出存在的SPF节点的列表在文档中。

文档中第一个网络需要用“Network #1”来定义，第二个则是” Network #2”等等（如样例）。每个SPF节点，输出一行，方式如下方例子所示，列出节点的编号和当这个节点失效时完全连通的子网络的个数。如果网络中没有SPF节点，输出“NO SPF nodes“即可。

**样例输入：**

1 2

5 4

3 1

3 2

3 4

3 5

0

1 2

2 3

3 4

4 5

5 1

0

1 2

2 3

3 4

4 6

6 3

2 5

5 1

0

0

**样例输出**

Network #1

SPF node 3 leaves 2 subnets

Network #2

No SPF nodes

Network #3

SPF node 2 leaves 2 subnets

SPF node 3 leaves 2 subnets

**读入是难点，接下来是找割点，删除割点并且DFS找点双连通分量**

**POJ3177**

**描述**

为了从F（1≤F≤5000）个牧场（从1到F进行编号）中的一个到达另一个牧场，Bessie和他的牛只能穿过烂苹果林。这些牛现在都累了，不想老走这个特殊的小路而是想要建一些新的小路，这样它们就有至少两种选择在任意牧场之间穿梭。在任意的两个牧场之间，它们现在至少有一条路径而他们想要至少有2条路径。

现在给出建好的R（F-1≤R≤10000）条路，每条都连接了两个不同的牧场，确定新修道路的最小数量（每个也是连接两个牧场）使得任意两个牧场之间都能有两条路连通。道路都是不同的，不会有相同的小路，即使他们在沿路的中间会碰到相同的牧场。

这里可能已经有一对牧场有超过一条小路的，并且你可能也会修一条新的另外的小路来连接相同的牧场。

**输入**

第一行：两个空格隔开的整数：F和R

第二到第R+1行：每行包括两个空格隔开的整数，表示一些小路端点所在的牧场。

**输出**

第一行：一个单独的整数，表述新修小道的数量

**样例输入**

7 7

1 2

2 3

3 4

2 5

4 5

5 6

5 7

**样例输出**

2

**提示**

例子的解释：

一个形象化的小路：

1 2 3  
 +---+---+   
 | |  
 | |  
 6 +---+---+ 4  
 / 5  
 /   
 /   
 7 +

建一个从1到6和从4到7的新路可以满足条件。

1 2 3  
 +---+---+   
 : | |  
 : | |  
 6 +---+---+ 4  
 / 5 :  
 / :  
 / :  
 7 + - - - -

检查一些路径：

1 – 2: 1 –> 2 and 1 –> 6 –> 5 –> 2   
1 – 4: 1 –> 2 –> 3 –> 4 and 1 –> 6 –> 5 –> 4   
3 – 7: 3 –> 4 –> 7 and 3 –> 2 –> 5 –> 7

可以看出每对牧场都是由两条路连接的

有可能加一些别的下路也能解决这个问题（比如6到7加一条道）。然而加两条已经是最小了。

**POJ3352**

**描述：**

夏天的建造时间到了。这一年，遥远热带小岛上的街道的管理员要修复并且升级岛上街道，这些街道连接了小岛上不同的景点。

这些街道很有意思。由于这个小岛的奇怪传统，这些街道需要被规划的没有任何十字路口，但是可以用桥和隧道来连通。在这种情况下，每条路就会在特定的景点之间连接，这样旅客就不会迷路了。

不幸的是，每条路都需要修理和升级，当建筑公司在一条特定的道路上工作时，这条道路是无法使用的。这就会导致一个问题，即使这个公司一次只修一条路，仍然有两个景点是无法访问的，。

所以，这个岛上的道路部门就决定召唤你的咨询服务来帮助他们解决这个问题。最后决定在景点之间建设新的道路。如果一条道路在建设中，这些新的道路能使任意两个景点之间仍然有路能够到达。你的任务就是找到这个新修道路的最小数量。

**输入**

第一行的输入包括两个正整数n和r，其中3≤n≤1000是景点的数量，2≤r≤1000是道路的数量。景点则是从1到n进行标号。每个接下来的r行会包括两个整数，v和w，中间空格隔开，表示一个街道连接了标号为v和w的景点。注意街道是双向的，并且任意一对景点之间会有至多一条路直接连接它们。当然，你也要保证在现在的这个配置下，任意两个景点是连通的。

**输出**

一行，包括一个整数，给出我们需要添加的街道的最小数量

**输入样例**

Sample Input 1

10 12

1 2

1 3

1 4

2 5

2 6

5 6

3 7

3 8

7 8

4 9

4 10

9 10

Sample Input 2

3 3

1 2

2 3

1 3

**输出样例**

Output for Sample Input 1

2

Output for Sample Input 2

0

**两道题完全相同，都是求有桥连通图变为双连通图所需要最少的边的个数**