



无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

# 无向图的连通性

河南省实验中学信息技术组

2026 年 01 月 11 日



# 无向图的连通性

## 无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 桥 (割边)
- 割点
- 双连通分量
- 缩点



# 割点与桥

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 对于图  $G$ ，删除其中的某一个顶点  $x$  及与  $x$  关联的边后， $G$  分裂成两个或两个以上不相连的子图，则称  $x$  为图  $G$  的**割点**。
- 对于图  $G$ ，删除其中的某一条边  $e$ ， $G$  分裂成两个不相连的子图，则称  $e$  为图  $G$  的**桥**或**割边**。
- 一般无向图 (不一定连通) 的割点与桥就是它各个连通块的割点与桥。
- 根据著名计算机科学家 Robert Taijan 的名字命名的 Tarjan 算法能在线性时间内求出无向图的割点与桥。



# 双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 若无向图不存在割点，则称它为点双连通图。
- 若无向图不存在桥，则称它为边双连通图。
- 无向图的极大点双连通子图被称为点双连通分量，简记为 v-DCC。
- 无向图的极大边双连通子图被称为边双连通分量，简记为 e-DCC。
- 点双连通分量和边双连通分量统称为双连通分量，简记为 DCC<sup>1</sup>。

---

<sup>1</sup>Double Connected Component.



# 双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 无向连通图是点双连通图，当且仅当满足下列两个条件之一：
  - 图的顶点数不超过 2。
  - 图中任意两点都同时包含在至少一个简单环中。
- 无向连通图是边双连通图，当且仅当任意一条边都包含在至少一个简单环中。



# 算法前置概念

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

在讨论割点与桥的判定算法之前，Tarjan 算法有如下若干概念：

- 时间戳：对图进行 DFS 的过程中，按照顶点第一次被访问的时间顺序，依次给予  $n$  个顶点标记为  $1 \sim n$ ，该标记被称为时间戳，记为  $dfn[x]$ 。
- 追溯值：设  $subtree(x)$  表示搜索树中以  $x$  为根的子树，定义追溯值  $low[x]$  为以下顶点的时间戳的最小值<sup>2</sup>：
  - ①  $subtree(x)$  中的顶点。
  - ② 通过 1 条不在搜索树上的边，能够到达  $subtree(x)$  的顶点。

---

<sup>2</sup>顶点  $x$  能到达的最小时戳。



# 追溯值

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

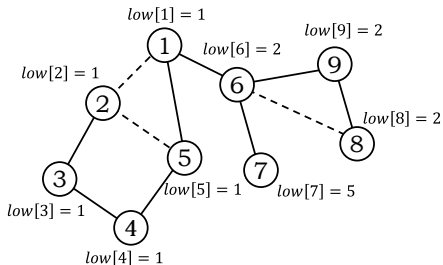
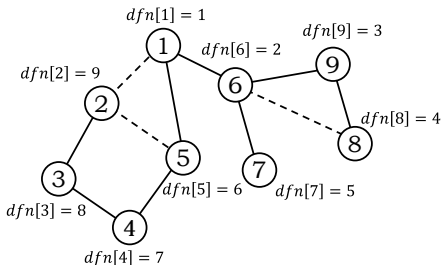
Network

练习

根据定义，为了计算  $low[x]$ ，则应当先令  $low[x] = dfn[x]$ ，然后考虑从  $x$  出发的每条边  $(x, y)$ ：

- 若  $(x, y)$  在搜索树上且  $x$  是  $y$  的父结点，则令  $low[x] = \min(low[x], low[y])$ 。
- 若  $(x, y)$  不在搜索树上，则令  $low[x] = \min(low[x], dfn[y])$ 。

我们以下图为例，求解各顶点的时间戳和追溯值，其中实线为搜索树上的边，虚线为非搜索树上的边。





# 追溯值

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 从顶点  $x$  出发
2 // 上次经过第  $last$  条边, 用于判定是否是搜索树上的边
3 // 为什么不用  $tarjan(int\ x, int\ fa)$ ?
4 void tarjan(int x, int last)
5 {
6     dfn[x] = low[x] = ++cnt;
7     for(int i = head[x]; i; i = nxt[i])
8     {
9         int y = ver[i];
10        if(!dfn[y]) // 未访问过的顶点  $y$ , 说明其在  $subtree(x)$  中
11        {
12            tarjan(y, i);
13            low[x] = min(low[x], low[y]);
14        }
15        else if(i != (last ^ 1)) // 边  $(x, y)$  不在搜索树上
16        {
17            low[x] = min(low[x], dfn[y]);
18        }
19    }
20 }
```



# 追溯值

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

点双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

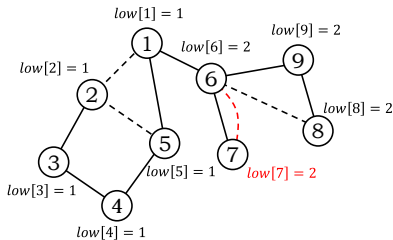
冗余路径

BLO

Network

练习

- 因为图为无向图，所以从顶点  $x$  出发，可以访问其父结点  $fa$ ，而根据  $low[x]$  的计算方法， $(x, fa)$  属于搜索树上的边，且  $fa$  不是  $x$  的子结点，故不能用  $dfn[fa]$  的来更新  $low[x]$ 。
- 但是如果利用父结点来判定边是否为生成树的边，会无法处理重边的情况。即当  $x$  与  $fa$  之间有重边时，重边显然不在搜索树上，那显然可以用  $dfn[fa]$  来更新  $low[x]$ 。



- 所以不能使用从顶点  $x$  到父结点  $fa$  的边，那么在利用邻接表存储时应当将边信息成对存储在 2 和 3、4 和 5、6 和 7 等位置，那么沿着第  $i$  条边  $(fa, x)$  到达  $x$ ，那么从  $x$  出发到达  $fa$  的边  $(x, fa)$  为  $i \text{ xor } 1$ ，忽略该边即可。



# 桥的判定

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

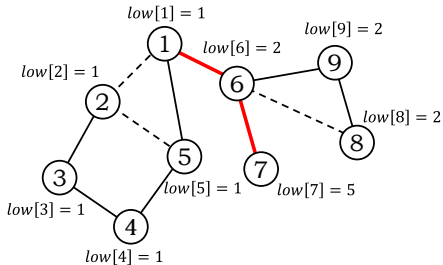
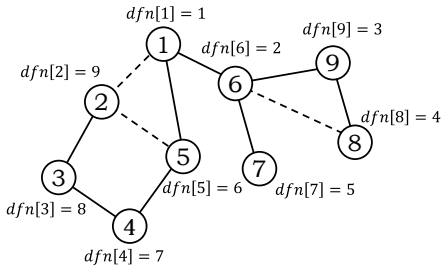
Network

练习

无向边  $(x, y)$  是桥，当且仅当搜索树上存在  $x$  的一个子结点  $y$ ，满足：

$$dfn[x] < low[y]$$

根据定义， $dfn[x] < low[y]$  说明从  $subtree(y)$  出发，在不经过  $(x, y)$  的前提下，无法到达  $x$  或比  $x$  更早访问的顶点。





# 桥的判定

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

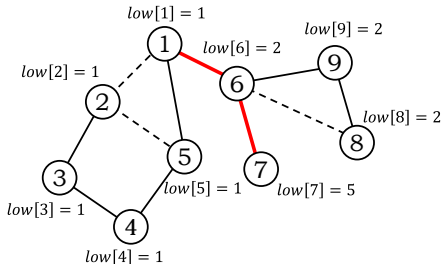
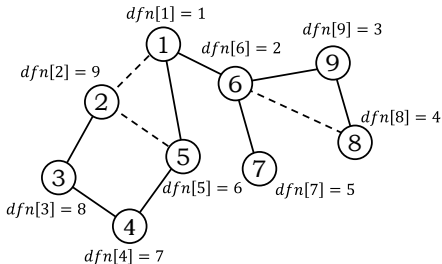
冗余路径

BLO

Network

练习

- 若不存在这样的子结点  $y$ , 使得  $dfn[x] < low[y]$ , 则说明每个  $subtree(y)$  都能通过其他边到达  $x$  或比  $x$  更早访问的顶点, 那说明  $(x, y)$  不是桥。
- 通过观察可以发现, 桥一定是搜索树中的边, 并且一个简单环中的边一定不是桥。





# 桥的判定

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

点双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 从顶点  $x$  出发, 上次经过第  $last$  条边
2 void tarjan(int x, int last)
3 {
4     dfn[x] = low[x] = ++cnt;
5     for(int i = head[x]; i; i = nxt[i])
6     {
7         int y = ver[i];
8         if(!dfn[y]) // 未访问过的顶点  $y$ , 说明其在  $subtree(x)$  中
9         {
10             tarjan(y, i);
11             low[x] = min(low[x], low[y]);
12             if(dfn[x] < low[y]) bridge[i] = bridge[i ^ 1] = true; // 割边
13         }
14         else if(i != (last ^ 1)) // 边  $(x, y)$  不在搜索树上
15         {
16             low[x] = min(low[x], dfn[y]);
17         }
18     }
19 }
```



# 割点判定

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

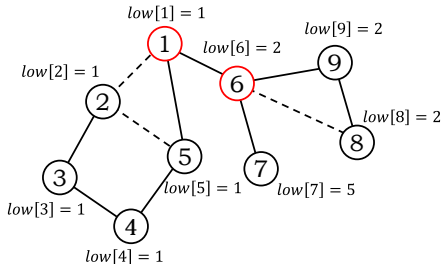
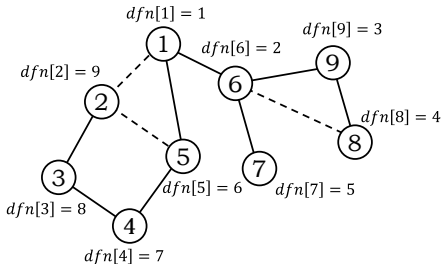
Network

练习

若  $x$  不是搜索树的根结点，则  $x$  是割点当且仅当存在  $x$  的一个子结点  $y$ ，满足：

$$dfn[x] \leq low[y]$$

特别地，若  $x$  是搜索树的根结点，则  $x$  是割点当前仅当  $x$  存在至少两个子结点  $y_1, y_2$  满足上述条件。





# 割点判定

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 从顶点  $x$  出发, 上次经过第  $last$  条边
2 void tarjan(int x, int last)
3 {
4     dfn[x] = low[x] = ++cnt;
5     int flag = 0; // 满足条件的子结点数
6     for(int i = head[x]; i; i = nxt[i])
7     {
8         int y = ver[i];
9         if(!dfn[y]) // 未访问过的顶点  $y$ , 说明其在  $subtree(x)$  中
10        {
11            tarjan(y, i);
12            low[x] = min(low[x], low[y]);
13            if(dfn[x] <= low[y]) // 割点
14            {
15                ++flag;
16                if(x != root || flag > 1) cut[x] = true;
17            }
18        }
19        else if(i != (last ^ 1)) // 边  $(x, y)$  不在搜索树上
20        {
21            low[x] = min(low[x], dfn[y]);
22        }
23    }
24 }
```



# 割点判定

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

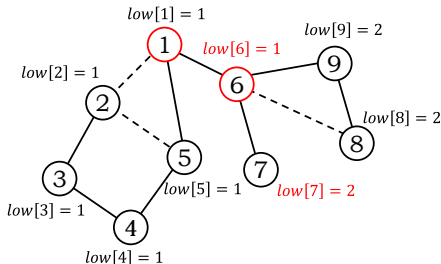
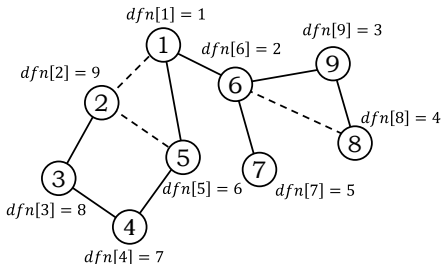
冗余路径

BLO

Network

练习

- 因为割点的判定法则是删除  $x$  及与  $x$  关联的所有边后，图分裂成两个或两个以上不相连的子图，所以在求割点时，不必考虑父结点和重边问题，从  $x$  出发能访问到的所有顶点的时间戳都能用来更新  $low[x]$ 。
- 那么图的存储和访问方式将不会有限制，但是没有太大意义。





## 割点判定 (不用考虑父结点和重边)

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 从顶点 x 出发
2 void tarjan(int x)
3 {
4     dfn[x] = low[x] = ++cnt;
5     int flag = 0; //满足条件的子结点数
6     for(int i = head[x]; i; i = nxt[i])
7     {
8         int y = ver[i];
9         if(!dfn[y]) // 未访问过的顶点 y, 说明其在 subtree(x) 中
10        {
11            tarjan(y);
12            low[x] = min(low[x], low[y]);
13            if(dfn[x] <= low[y]) // 割点
14            {
15                ++flag;
16                if(x != root || flag > 1) cut[x] = true;
17            }
18        }
19        else low[x] = min(low[x], dfn[y]); // 边 (x,y) 不在搜索树上 (不需要考虑父结点)
20    }
21 }
```



## 边双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

缩点

点双连通分量

缩点

例题

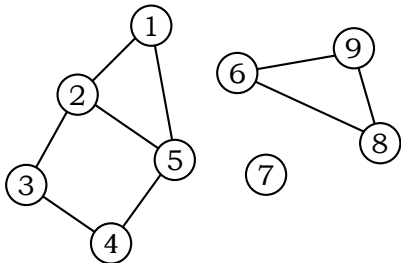
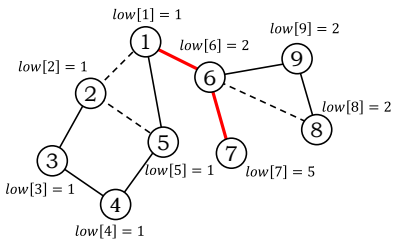
冗余路径

BLO

Network

练习

- 求出无向图中所有的桥，把桥删除后，无向图会分为若干个连通块，每一个连通块就是一个边双连通分量。
- 先利用 Tarjan 算法求出所有的桥，然后再进行 DFS(不经过桥)，划分出连通块。





# 边双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 int dcc = 0; // 边双连通分量标记
2
3 void dfs(int x)
4 {
5     c[x] = dcc;
6     for(int i = head[x]; i; i = nxt[i])
7     {
8         int y = ver[i];
9         if(c[y] || bridge[i]) continue;
10        dfs(y);
11    }
12 }
13
14 for(int i = 1; i <= n; ++i)
15 {
16     if(!c[i]) ++dcc, dfs(i);
17 }
```



## 边双连通分量缩点

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

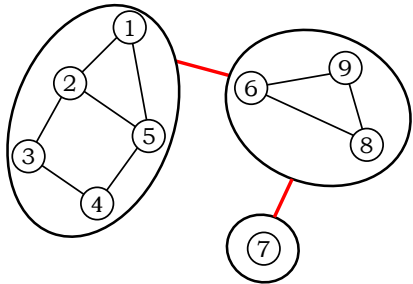
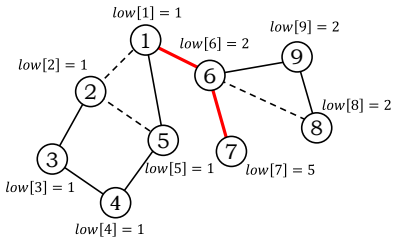
冗余路径

BLO

Network

练习

- 把每个边双连通分量看作一个顶点，把桥  $(x, y)$  看作连接编号为  $c[x]$  和  $c[y]$  的边双连通分量对应顶点的无向边，那么将会产生一棵树 (如果原图不连通，则产生森林)。
- 这种将边双连通分量收缩为一个顶点的方法称为**缩点**。





# 边双连通分量缩点

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 遍历原图的所有边
2 for(int i = 2; i <= tot; i += 2)
3 {
4     int x = ver[i ^ 1], y = ver[i]; // 边 (x,y)
5     if(c[x] == c[y]) continue; // 如果在同一个边双连通分量中，则不需要处理
6     add(c[x], c[y]), add(c[y], c[x]); // 建边 (c[x],c[y])
7 }
```



# 点双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

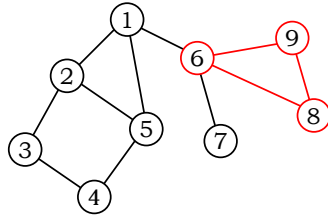
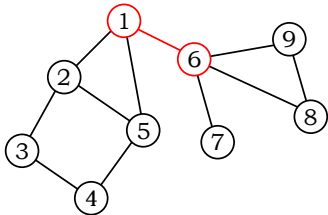
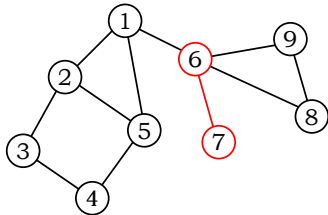
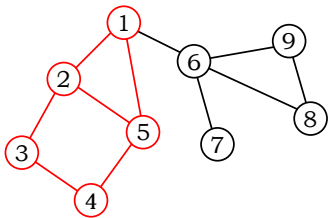
冗余路径

BLO

Network

练习

- 若某个顶点为孤立点，则它自己单独构成一个点双连通分量。除此之外，点双连通分量的大小至少为 2。
- 注意，和普通连通分量不重合不同，割点可能属于多个点双连通分量。





# 点双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

为了求出点双连通分量，需要在 Tarjan 算法的过程中维护一个栈，并按照如下方法维护栈中的元素：

- ① 当一个顶点第一次被访问时，把该顶点入栈。
- ② 当割点判定法则的条件  $dfn[x] \leq low[y]$  成立时，无论  $x$  是否为根，都要：
  - ① 从栈顶不断弹出所有顶点，直至顶点  $y$  被弹出。
  - ② 刚才弹出的所有顶点与顶点  $x$  构成一个点双连通分量。



# 点双连通分量

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 void tarjan(int x, int last)
2 {
3     dfn[x] = low[x] = ++cnt;
4     if(x == root && head[x] == 0) { vdcc[++dcc].push_back(x); return; } // 孤立点
5     st.push(x);
6     int flag = 0; // 满足条件的子结点数
7     for(int i = head[x]; i; i = nxt[i]) {
8         int y = ver[i];
9         if(!dfn[y]) { // 未访问过的顶点 y, 说明其在 subtree(x) 中
10             tarjan(y, i);
11             low[x] = min(low[x], low[y]);
12             if(dfn[x] <= low[y]) { // 割点
13                 ++flag;
14                 if(x != root || flag > 1) cut[x] = true;
15                 ++dcc;
16                 for(;;) { // 从栈顶不断弹出顶点, 直到顶点 y 被弹出
17                     int z = st.top(); st.pop();
18                     vdcc[dcc].push_back(z);
19                     if(z == y) break;
20                 }
21                 vdcc[dcc].push_back(x);
22             }
23         }
24         else if(i != (last ^ 1)) low[x] = min(low[x], dfn[y]); // 边 (x, y) 不在搜索树上
25     }
26 }
```



# 点双连通分量缩点

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

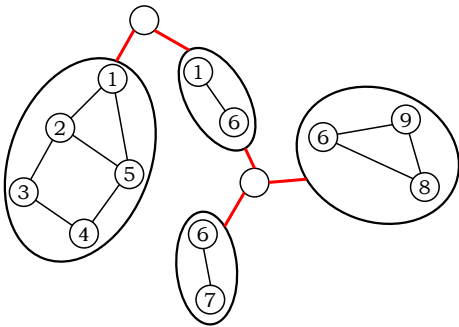
冗余路径

BLO

Network

练习

- 在点连通分量中，一个割点可能存在与多个点连通分量。
- 设图中有  $p$  个割点和  $t$  个点连通分量，那么我们建立一张包含  $p + t$  个顶点的新图，把每个点连通分量和每个割点都作为新图中的顶点，并在每个割点与包含它的所有点连通分量之间连边。





# 点双连通分量缩点

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // 给每个割点一个新的编号
2 int num = dcc;
3 for(int i = 1; i <= n; ++i)
4 {
5     if(cut[i]) nid[i] = ++num;
6 }
7 // 建图，每个点连通分量到它包含的所有割点连边
8 for(int i = 1; i <= dcc; ++i)
9 {
10     for(int j = 0; j < vdcc[i].size(); ++j)
11     {
12         int x = vdcc[i][j];
13         if(cut[x]) add(i, nid[x]), add(nid[x], i);
14         else c[x] = i; // 除割点外，其他点仅属于一个点连通分量
15     }
16 }
```



## 【例】冗余路径

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

为了从  $F$  ( $1 \leq F \leq 5,000$ ) 个牧场 (编号为 1 到  $F$ ) 中的一个到达另一个牧场, 贝西和其他牛群被迫经过腐烂苹果树附近。奶牛们厌倦了经常被迫走特定的路径, 想要修建一些新路径, 以便在任意一对牧场之间总是有至少两条独立的路线可供选择。目前在每对牧场之间至少有一条路径, 他们希望至少要有两条。当然, 他们只能在官方路径上从一个牧场移动到另一个牧场。

给定当前  $R$  ( $F - 1 \leq R \leq 10,000$ ) 条路径的描述, 每条路径恰好连接两个不同的牧场, 确定必须修建的最少新路径数量 (每条新路径也恰好连接两个牧场), 以便在任意一对牧场之间至少要有两条独立的路线。若两条路线不使用相同的路径, 即使它们沿途访问相同的中间牧场, 也被视为独立的。

在同一对牧场之间可能已经有多条路径, 你也可以修建一条新路径连接与某条现有路径相同的牧场。



## 【例】冗余路径

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

### 【输入格式】

第 1 行：两个用空格分隔的整数： $F$  和  $R$ 。

第 2 行到第  $R + 1$  行：每行包含两个用空格分隔的整数，表示某条路径的两个端点牧场。

### 【输出格式】

第 1 行：一个整数，表示必须修建的新路径数量。

### 【样例 1 输入】

```
7 7
1 2
2 3
3 4
2 5
4 5
5 6
5 7
```

### 【样例 1 输出】

```
2
```



## 【例】冗余路径

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 问题转化为：最少需要加多少条无向边使得图转化为边双连通图。
- 图不是边双连通的主要原因是存在割边，所以原图非割边对答案没有影响。
- 求出原图所有割边，然后缩点，在缩点产生的树上连边使得所有的割边都至少被覆盖一次。
- 为了最大化覆盖割边，将缩点后产生的树的叶子结点两两连边即可，叶子结点个数为奇数时，将剩余的叶子结点和任意点连边即可。
- 设缩点后树的叶子结点的个数为  $k$ ，那么答案为  $\lceil \frac{k}{2} \rceil$ 。



## 【例】冗余路径

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 // d[x] 为 x 的度
2 tarjan(1, 0);
3 for(int i = 1; i <= n; ++i) if(!c[i]) ++dcc, dfs(i);
4 for(int i = 2; i <= tot; i += 2)
5 {
6     int x = ver[i], y = ver[i ^ 1];
7     if(c[x] == c[y]) continue;
8     ++d[c[x]], ++d[c[y]];
9 }
10 int ans = 0;
11 for(int i = 1; i <= dcc; ++i) ans += (d[i] == 1);
12 printf("%d", (ans + 1) / 2);
```



## 【例】BLO-Blockade

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

B 城有  $n$  个城镇（从 1 到  $n$  标号）和  $m$  条双向道路。

每条道路连结两个不同的城镇，没有重复的道路，所有城镇连通。

把城镇看作节点，把道路看作边，容易发现，整个城市构成了一个无向图。

请你对于每个节点  $i$  求出，把与节点  $i$  关联的所有边去掉以后（不去掉节点  $i$  本身），无向图有多少个有序点  $(x, y)$ ，满足  $x$  和  $y$  不连通。



## 【例】BLO-Blockade

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

### 【输入格式】

第一行包含两个整数  $n$  ( $n \leq 10^5$ ) 和  $m$  ( $m \leq 5 \times 10^5$ )。

接下来  $m$  行，每行包含两个整数  $a$  和  $b$ ，表示城镇  $a$  和  $b$  之间存在一条道路。

### 【输出格式】

输出共  $n$  行，每行输出一个整数。

第  $i$  行输出的整数表示把与节点  $i$  关联的所有边去掉以后（不去掉节点  $i$  本身），无向图有多少个有序点  $(x, y)$ ，满足  $x$  和  $y$  不连通。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
8
8
16
14
8
```



## 【例】BLO-Blockade

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 如果  $x$  不是割点，那么删除  $x$  的所有边后， $x$  自己成为一个连通块，其他  $n - 1$  个顶点构成一个连通块，故而答案为  $2(n - 1)$ 。
- 如果  $x$  是割点，那么删除  $x$  的所有边后，图会分成若干个连通块。求出每个连通块大小后，答案为连通块大小两两相乘再相加，但是实际处理比较麻烦。
- 在图的连通性问题中，习惯从搜索树的角度分析，对于  $x$ ，它的子结点  $y_i$  满足  $dfn[x] \leq low[y_i]$ ，那么删除  $x$  的所有边后：
  - 点  $x$  自身构成一个连通块；
  - 以  $y_i$  为根的子树中的结点构成一个连通块；
  - 其他结点构成一个连通块。



## 【例】BLO-Blockade

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 定义  $d[x]$  表示以  $x$  为根的子树大小。
- 对于  $x$ ，它的子结点  $y_i$  满足  $dfn[x] \leq low[y_i]$ ，那么删除  $x$  的所有边后，不连通的有序对数量为：
  - 以  $y_i$  为根的子树与其他点 (包括  $x$ ) 构成的有序对数量  $\sum d[y_i] \times (n - d[y_i])$ ；
  - 点  $x$  与其他点构成的有序对  $n - 1$ 。
  - 除了  $x$  和满足条件的以  $y_i$  为根的子树与  $x$  和满足条件的以  $y_i$  为根的子树构成的有序对数量  $(n - 1 - \sum d[y_i]) \times (1 + \sum d[y_i])$ 。
- 其中第①项可以在 Tarjan 算法遍历子结点的过程中计算， $\sum d[y_i]$  也可以在遍历子结点时累加得出，那么后两项在遍历子结点后计算即可。



## 【例】BLO-Blockade

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 void tarjan(int x) {
2     d[x] = 1;
3     dfn[x] = low[x] = ++cnt;
4     int flag = 0, sum = 0; // 子树结点数
5     for(int i = head[x]; i; i = nxt[i]) {
6         int y = ver[i];
7         if(!dfn[y]) {
8             tarjan(y);
9             d[x] += d[y];
10            low[x] = min(low[x], low[y]);
11            if(dfn[x] <= low[y])
12            {
13                sum += d[y];
14                ans[x] += (long long) d[y] * (n - d[y]);
15                ++flag;
16                if(x != 1 || flag > 1) c[x] = 1;
17            }
18        }
19        else low[x] = min(low[x], dfn[y]);
20    }
21    if(c[x]) ans[x] += (n - 1) + (long long)(n - sum - 1) * (sum + 1);
22    else ans[x] = 2 * (n - 1);
23 }
```



## 【例】Network

网络管理员管理大型网络。网络由  $N$  台计算机和  $M$  对计算机之间的链路组成。任何一对计算机都是通过连续链接直接或间接连接的，因此数据可以在任意两台计算机之间进行转换。管理员发现某些链接对网络至关重要，因为其中任何一个链接的失败都可能导致某些计算机之间的数据无法转换。他把这种联系称为桥梁。他计划一个接一个地增加一些新的链接，以消除所有的桥梁。

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习



## 【例】Network

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

### 【输入格式】

输入包含多组测试数据。

每组测试数据，第一行包含两个整数  $N$  ( $N \leq 10^5$ ) 和  $M$  ( $N - 1 \leq M \leq 2 \times 10^5$ )。接下来  $M$  行，每行包含两个整数  $A$  和  $B$ ，表示点  $A$  和点  $B$  之间有一条边，点的编号为  $1 \sim N$ 。

接下来一行，包含整数  $Q$  ( $1 \leq Q \leq 1000$ )。

在接下来  $Q$  行，每行包含两个整数  $A$  和  $B$ ，表示在  $A$  和  $B$  之间加一条边后剩余的桥梁数。

当输入 0 0 时表示输入终止。

### 【输出格式】

每组数据第一行输出 Case  $x$ :，其中  $x$  为组别编号，从 1 开始。

接下来  $Q$  行，每行输出一个整数，表示一次询问的结果。

每组数据输出完毕后，输出一个空行。



## 【例】Network

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

### 【样例 1 输入】

```
3 2
1 2
2 3
2
1 2
1 3
4 4
1 2
2 1
2 3
1 4
2
1 2
3 4
0 0
```

### 【样例 1 输出】

```
Case 1:
1
0

Case 2:
2
0
```



## 【例】Network

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 每次向图中增加一条边，可能会覆盖原图中的某些桥，然后询问剩余的桥的数量。
- 先利用 Trajan 算法求出所有的边双连通分量，然后缩点，得到一棵树。
- 最初，剩余的桥的数量为树的总边数。
- 对于每个添加边  $(x, y)$ ，如果它们在同一个边双连通分量，那么桥的数量不变。
- 如果它们在不同的边双连通分量，设它们所在的边双连通分量分别为  $c[x], c[y]$ ，那么从  $c[x] \rightarrow \text{lca}(c[x], c[y]) \rightarrow c[y]$  上的所有的桥都被覆盖，对这些被覆盖的桥做标记并在总数量上减去这些桥即可。
- 时间复杂度： $O(M + QN)$ 。



## 【例】Network

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 if(c[x] != c[y])
2 {
3     x = c[x], y = c[y];
4     if(d[x] < d[y]) swap(x, y);
5     for(int i = d[x]; i > d[y]; --i)
6     {
7         if(!cover[x]) cover[x] = 1, --ans;
8         x = p[x];
9     }
10    while(x != y)
11    {
12        if(!cover[x]) cover[x] = 1, --ans;
13        if(!cover[y]) cover[y] = 1, --ans;
14        x = p[x], y = p[y];
15    }
16 }
```



## 【例】Network

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 并查集优化
- 在上述从  $c[x], c[y]$  到  $\text{lca}(c[x], c[y])$  的过程中，如果某个桥被覆盖，那么将这条边的子结点所在的集合合并到父结点所在的集合。
- 在  $c[x], c[y]$  到  $\text{lca}(c[x], c[y])$ ，可以直接跳到并查集的代表结点。
- 时间复杂度： $O(M + Q \log N)$ 。



## 【例】Network

无向图的连通性

河南省实验中学信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

```
1 if(c[x] != c[y])
2 {
3     x = c[x], y = c[y];
4     int z = lca(c[x], c[y]);
5     while(d[x] > d[z])
6     {
7         if(x == get(x)) --ans, fa[x] = f[x][0];
8         x = get(x);
9     }
10    while(d[y] > d[z])
11    {
12        if(y == get(y)) --ans, fa[y] = f[y][0];
13        y = get(y);
14    }
15 }
```



# 练习

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 炸铁路 (洛谷 P1656)
- 割点 (洛谷 P3388)
- 备用交换机 (COGS 8)
- BLO-Blockade(COGS 3143)



# 练习

无向图的连通性

河南省实验中学  
信息技术组

概念

算法前置

桥的判定

割点判定

边双连通分量

边双连通分量

缩点

点双连通分量

点双连通分量

缩点

例题

冗余路径

BLO

Network

练习

- 冗余路径 (COGS 311)
- Network(COGS 3157)
- 多余的路径 (CF1000E)
- 矿场搭建 [HNOI 2012](COGS 1348)
- 铁人两项 [APIO2018](洛谷 P4630)
- 圆桌骑士 (COGS 3250)
- 逃不掉的路 (COGS 3641)
- 交通实时查询系统 (COGS 3642)
- 战略游戏 [SDOI2018](洛谷 P4606)
- 建造军营 [NOIP 2022](COGS 3813)