

# 点双连通分量的求解



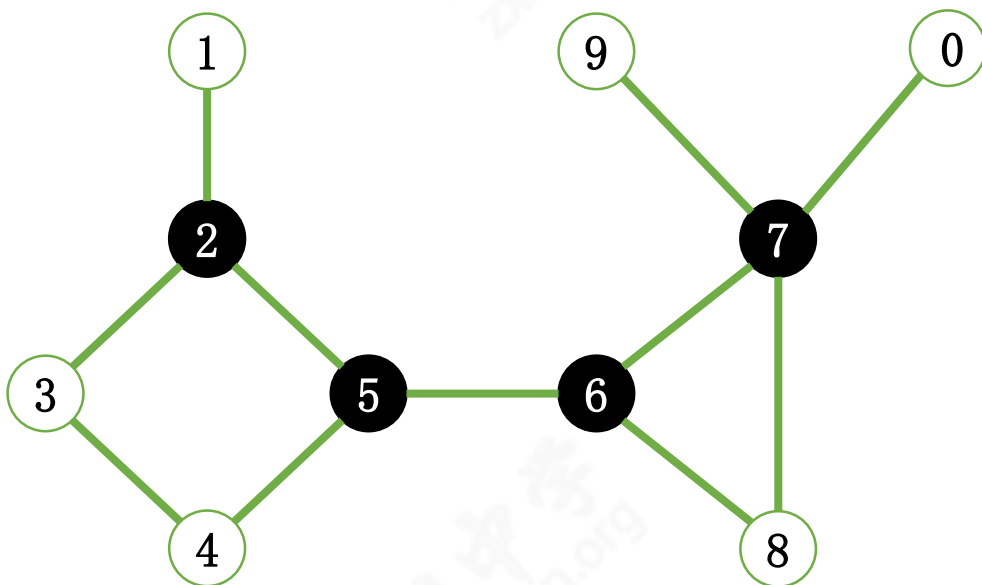
主讲人：邓哲也



# 点双连通分量的求解

首先可以用 Tarjan 算法求出图中的全部割点。

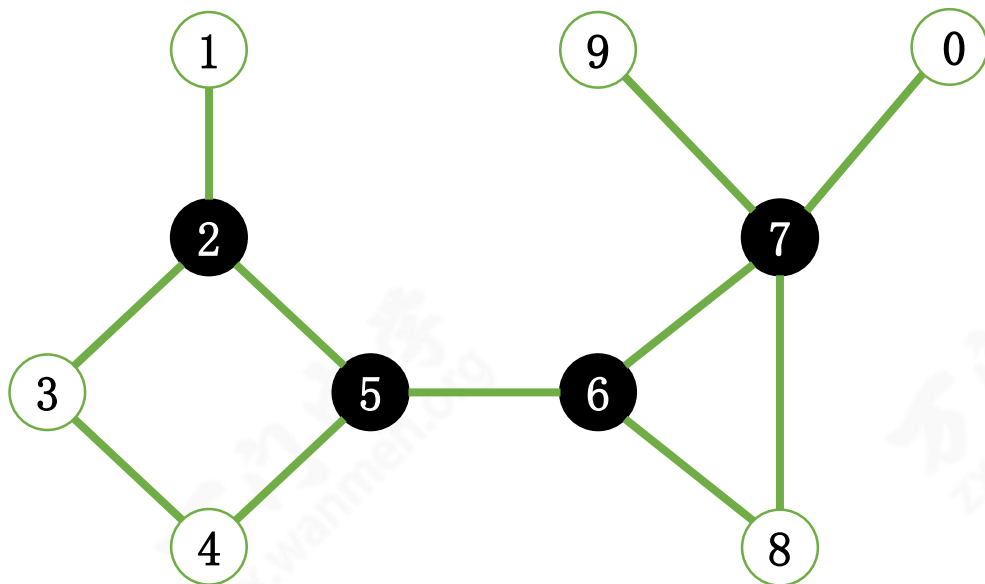
但是一个割点可能属于多个连通分量。



# 点双连通分量的求解

在 DFS 过程中，每找到一条生成树的边，就把这条边加入栈中。

如果遇到  $low[v] \geq dfn[u]$ ，说明  $u$  是一个割点，同时把边从栈顶一条条取出，直到遇到边  $(u, v)$ ，取出这些边与其关联的顶点，组成一个点双连通分量。

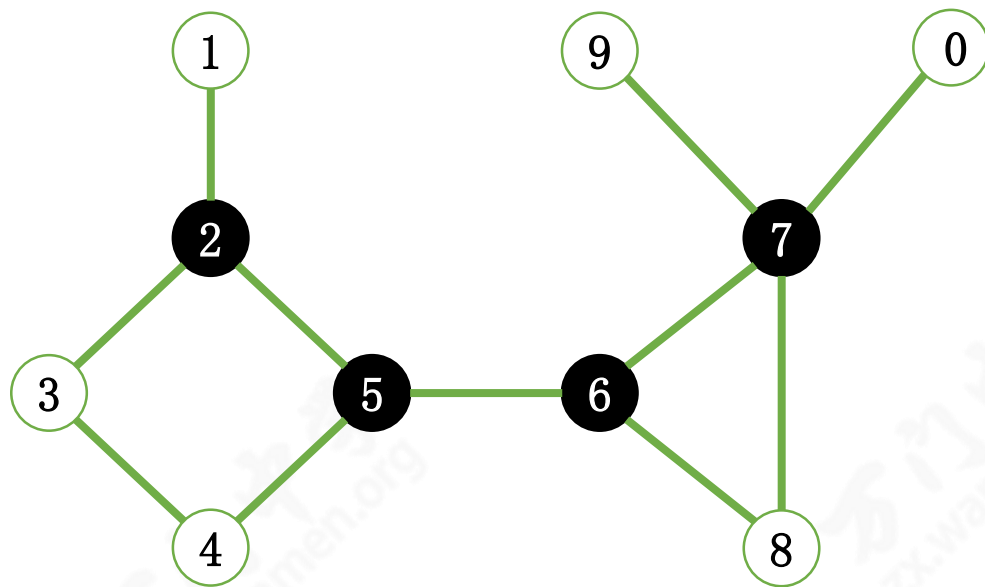


# 代码实现

建立一个栈存边：

```
pair <int, int> st[N];
```

```
int top;
```



# 代码实现

```
void tarjan(int u) {  
    vis[u] = 1;  
    dfn[u] = low[u] = ++ tot;  
    for (int i = h[u]; i != -1; i = e[i].next) {  
        int v = e[i].v;  
        if (!vis[v]) {  
            st[++ top] = make_pair(u, v);  
            tarjan(v);  
            low[u] = min(low[u], low[v]);  
        }  
    }  
}
```

# 代码实现

```
if (low[v] >= dfn[u]) {
    b[u] ++;
    while(1) {
        int uu = st[top].first, vv = st[top].second;
        top --;
        printf("%d %d\n", uu, vv);
        if((uu == u && vv == v) || (uu == v && vv == u))
            break;
    }
    printf("\n");
}
else {
    low[u] = min(low[u], dfn[v]);
}
}
```

# 代码实现

在这张图上调用 Tarjan(5):

输出:

7 10

7 9

7 8

6 7

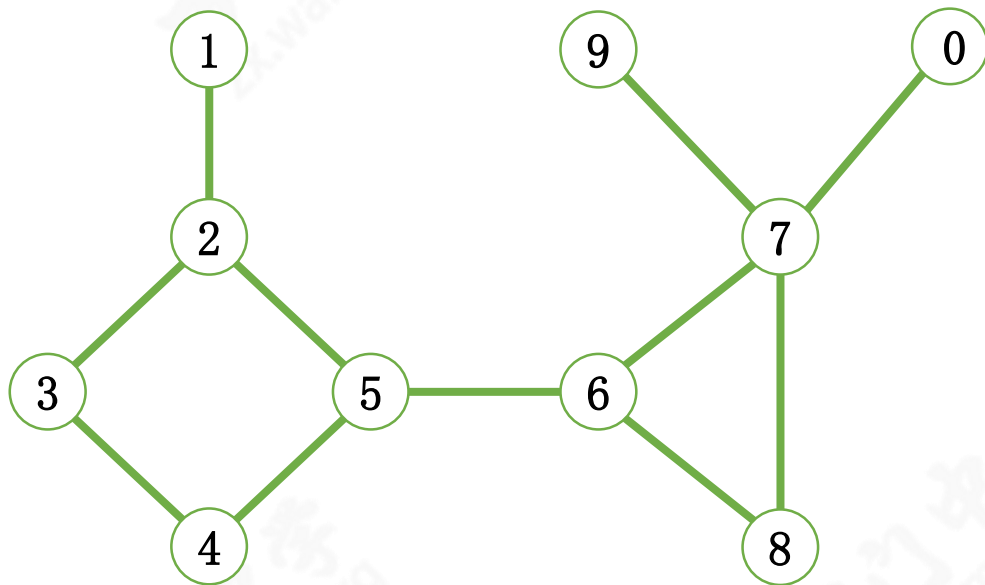
5 6

2 1

3 4

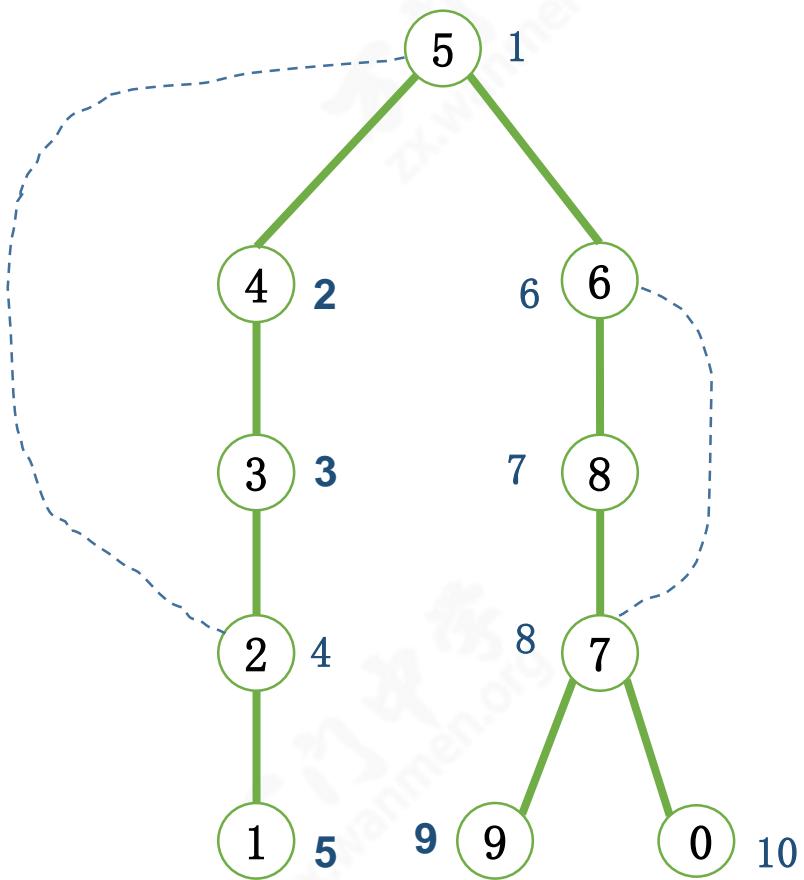
2 3

5 2



# 代码实现过程

调用 Tarjan(5)





下节课再见