Tarjan求割点代码实现

主讲人：邓哲也

代码实现

我们用邻接表来存边。

```
struct edge {
    int v, next;
}e[M];
int h[N], ee, dfn[N], low[N], tot, b[N], vis[N], n, m;

void addedge(int u, int v){
    e[ee] = (edge){v, h[u]};
    h[u] = ee ++;
}
```

代码实现

b[N] 数组用于统计去掉点 u， 原来的连通图分成了几个连通分量。

如果 u 是根节点，那么有几个子节点，就有几个连通分量。

如果 u 不是根节点，那么如果有几个子节点 v 满足 low[v] >= dfn[u]，就有 v + 1 个连通分量。

代码实现

Tarjan算法主体部分：

```
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]) {
            tarjan(v);
            low[u] = min(low[u], low[v]);
            if (low[v] >= dfn[u]) b[u] ++;
        } else {
            low[u] = min(low[u], dfn[v]);
        }
    }
}
```

代码实现

调用Tarjan算法：

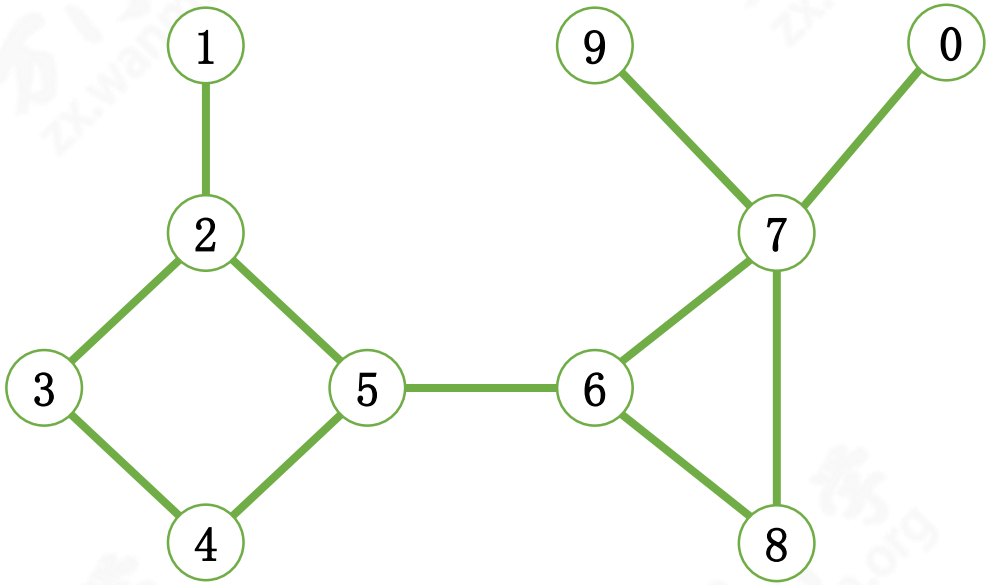
```
for(int i = 1;i <= n;i ++)
   if(!vis[i]){
        tarjan(i);
        b[i] --;
    }
for(int i = 1;i <= n;i ++)
    printf(“%d %d\n”, i, b[i] + 1);
```

代码实现

在这张图上调用 Tarjan 算法：

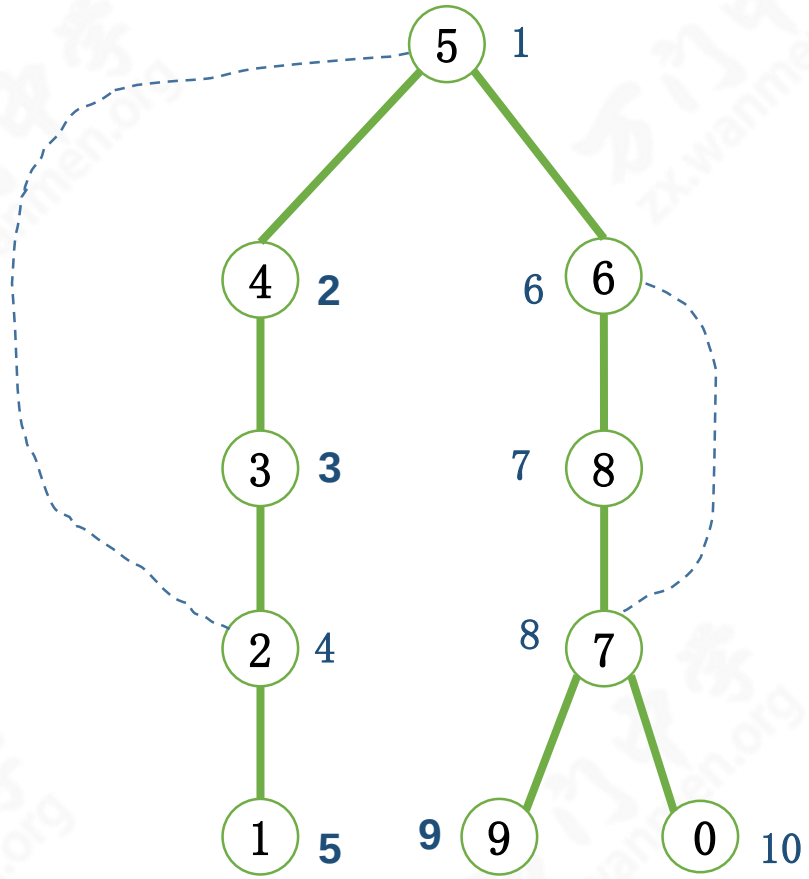
输出：

```
1 1
2 2
3 1
4 1
5 2
6 2
7 3
8 1
9 1
0 1
```



代码实现

调用 Tarjan(5)



	1	2	3	4	5	6	7	8	9	0
dfn	5	4	3	2	1	6	8	7	9	10
low	5	1	1	1	1	6	6	6	9	10
b	1	2	1	1	2	2	3	1	1	1

下节课再见