**实验报告**

一、算法思路

**首先看关节点和桥的定义：**

* 关节点：无向连通图中，如果删除某点后，图变成不连通，则称该点为关节点。
* 桥：无向连通图中，如果删除某边后，图变成不连通，则称该边为桥。

**根据以上定义，结合Tarjan算法，定义如下两个数组visited和low：**

* visited[u]是指对图进行深度优先搜索时，访问到u节点的时间。所以只有当visited[u]大于零时，u节点才被访问过。
* low[u] 表示u或者u的子树中能够通过非父子边追溯到的最早的节点的DFS开始时间

**在这两个数组的基础上，我们有如下判断关节点的方法：**

1. 若根节点V的子树个数大于1，则该节点一定为关节点。
2. 对于非根节点v，它是割点当且仅当它有某个儿子w,使得Low[w]>=Visited[v]。因为如果满足Low[w]>=Visited[v]，则 v有某个孩子w，其可到达的节点的最早开始时间大于等于其父亲的开始时间。即w一定不可能到达比v开始时间早的节点（深度优先搜索中比v先访问的节点），则删除节点v后，w一定不能与v之前的节点相连，则该图不为连通图。

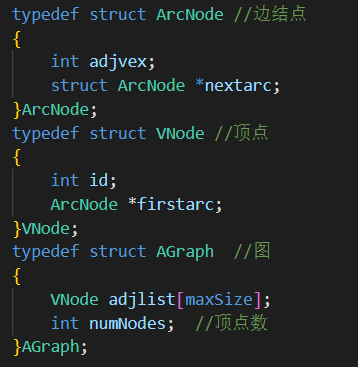
**结合课件的引理，有如下判断桥的方法：**

一条边(u,v)是桥，当且仅当(u,v)为树枝边(即非负边)，且满足visited[v] = Low[v]，因为这表示v 的子树中没有back edge指向较早发现的顶点，即产生桥。

二、代码说明

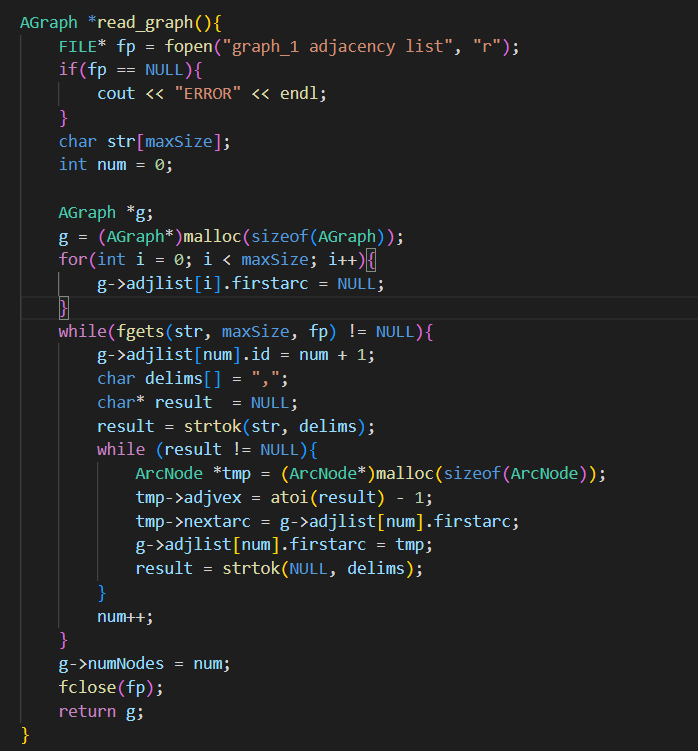
1. 数据结构

首先设置邻接表存储的图所需的数据结构

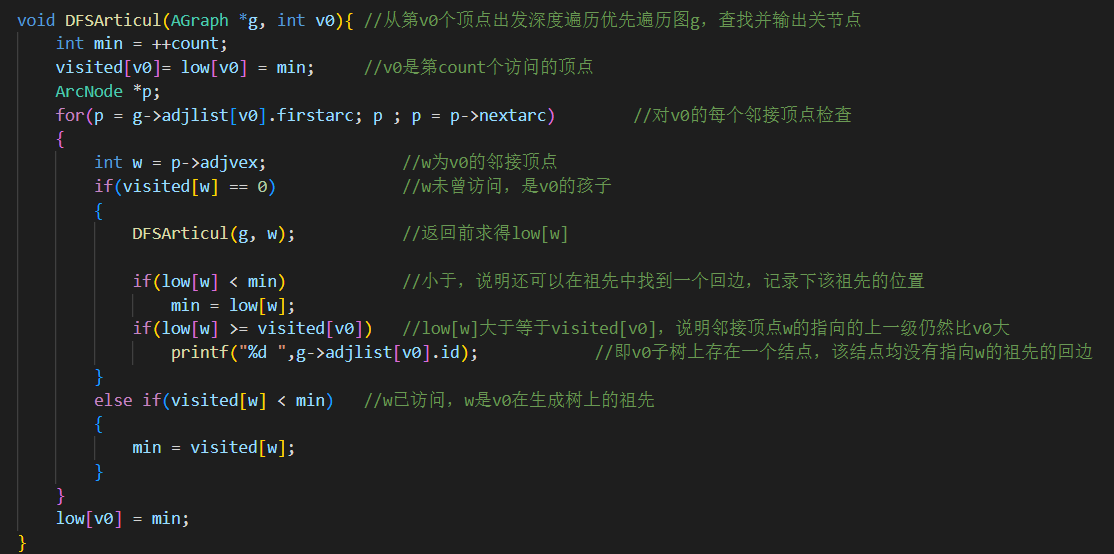


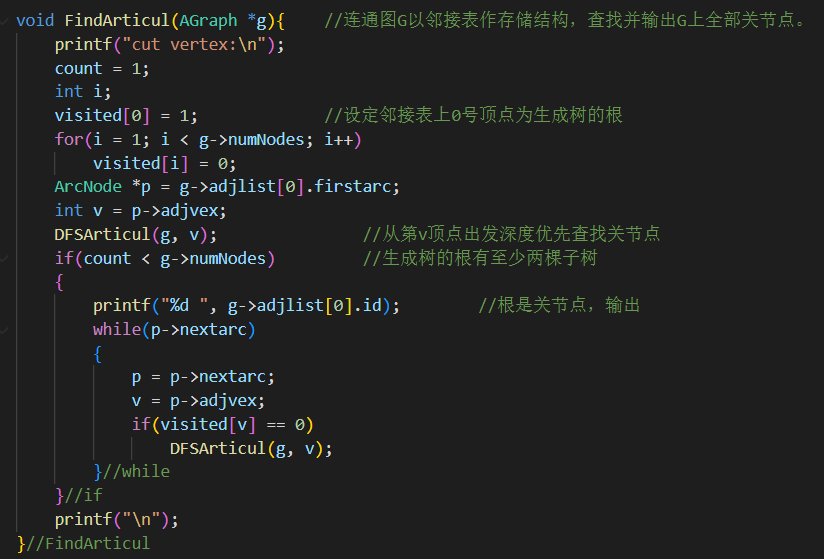
2. 初始化邻接表

read\_graph函数从对应文件中按行读取每个节点的邻接点，采用头插法填入构造的邻接表中。

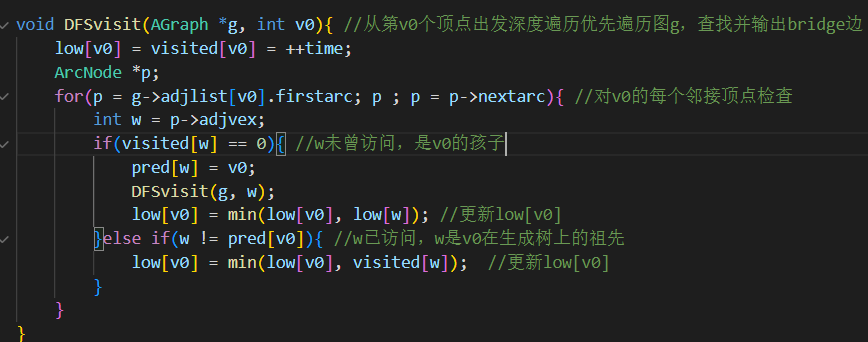


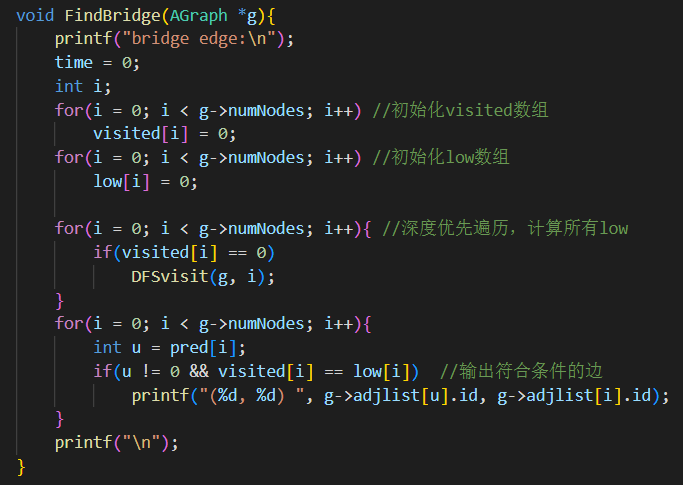
3. 求关节点





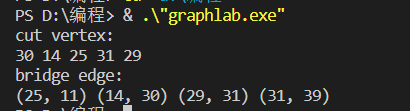
4. 求bridge 边





三、运行结果

对样例输入1：graph\_1 adjacency list，结果如下：



对样例输入2：graph\_2 adjacency list，结果如下：

