

P1

INPUT: directed graph G

OUTPUT: True OR False

```
# 找到结束时间最大的点
DFSALL(G)
max_finish = -1
max_node = null
for each node u in G:
    if u.f > max_finish:
        max_finish = u.f
        max_node = u
# 判断该点是否满足条件
num = n
Stack Q
Q.push(max_node)
while not Q.empty():
    x = Q.pop()
    if not x.visited:
        x.visited = True
        num -= 1
        for each edge (x, y) in E:
            Q.push(y)
# 返回结果
if num == 0:
    return max_node
else:
    return null
```

P2

(a)

```
def SOL(G):  
    # 拓扑排序, 返回节点序列  
    lst = TOPO-SORT(G)  
    # 判断第i个节点到第i+1个节点是否有边  
    for i = 0 to n-2:  
        x, y = lst[i], lst[i+1]  
        judge = False  
        for each node u in adj[x]: # adj是邻接矩阵  
            if u == y:  
                judge = True  
                break  
        if not judge:  
            return False  
    return True
```

(b)

```
# 计算G的无环分量图  
G_SCC = stronglt-connected-components(G)  
# 调用 (a) 中的方法  
SOL(G_SCC)
```