## P6 伪代码

```
# 初始化
for v in V:
    v.indegree = 0
for (u, v) in E:
    v.indegree += 1
# lst1是本轮删除的节点
# lst2是下一轮删除的节点
lst1 = []
lst2 = []
# ans为最小学期数
ans = 0
for v in V:
    if v.indegree == 0:
        lst1.add(v)
# 删除过程
while lst1 is not empty:
    ans += 1
    for u in lst1:
        for v in adj[u]: # adj是邻接表
            v.indegree -= 1
            if v.indegree == 0:
                lst2.add(v)
    lst1 = lst2
    lst2 = []
return ans
```

## P7伪代码

```
// 初始化
add each node from 1 to n^2 to V:
// adj是邻接链表
adj = []
for i in range(1, n^2 + 1):
    for j in range(1, k+1):
        if k is snake`s tail:
            x <- the snake`s head
            adj[i].add(x)
        elif k is ladder`s bottom:
            x <- the ladder`s top
            adj[i].add(x)
        else
            adj[i].add(k)
// BFS
return shortestLengthWithBFS(1, n^2)
```