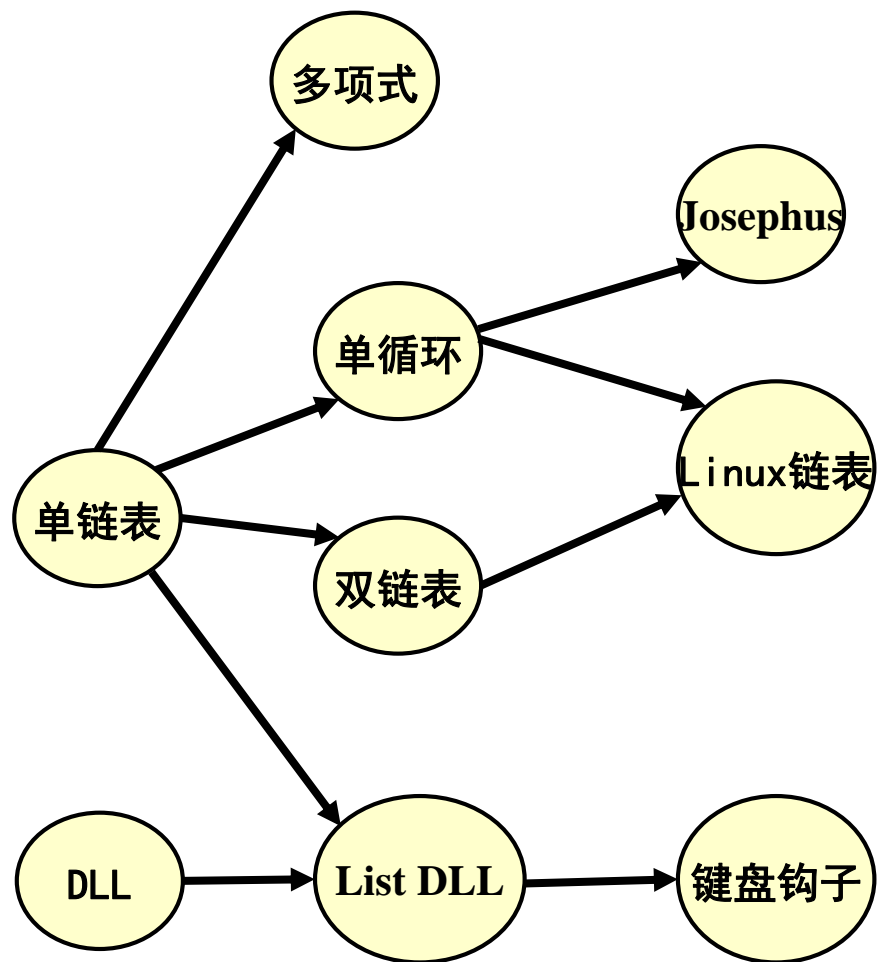


6.7 拓扑排序

AOV网: 顶点活动网(Activity On Vertex network)

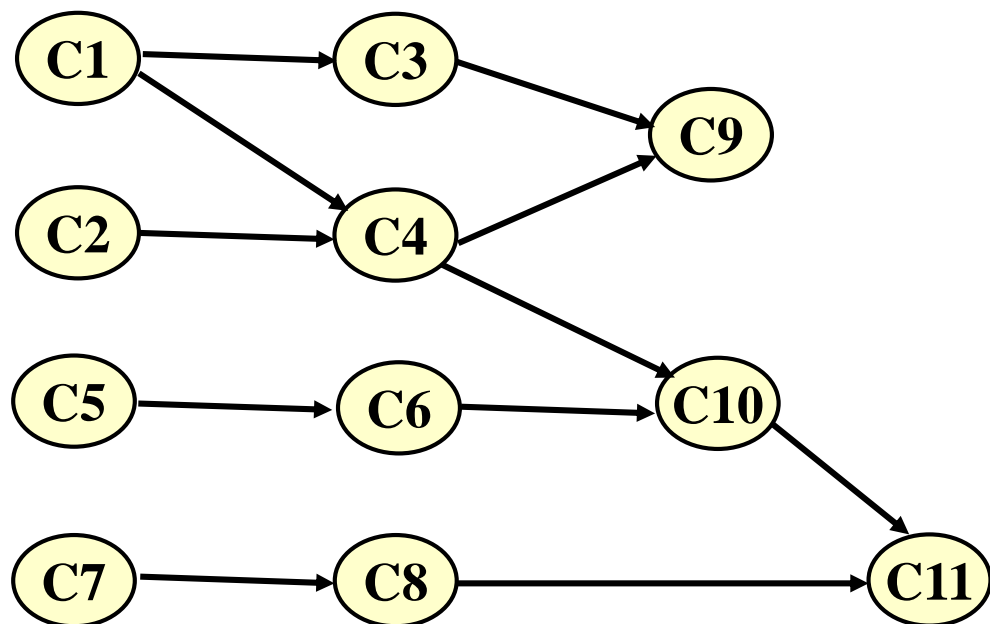


- 有向图;
- 无环图;
- 顶点表示活动;
- 边表示活动间的先后关系;

线性表视频知识点之间的依赖关系

返回

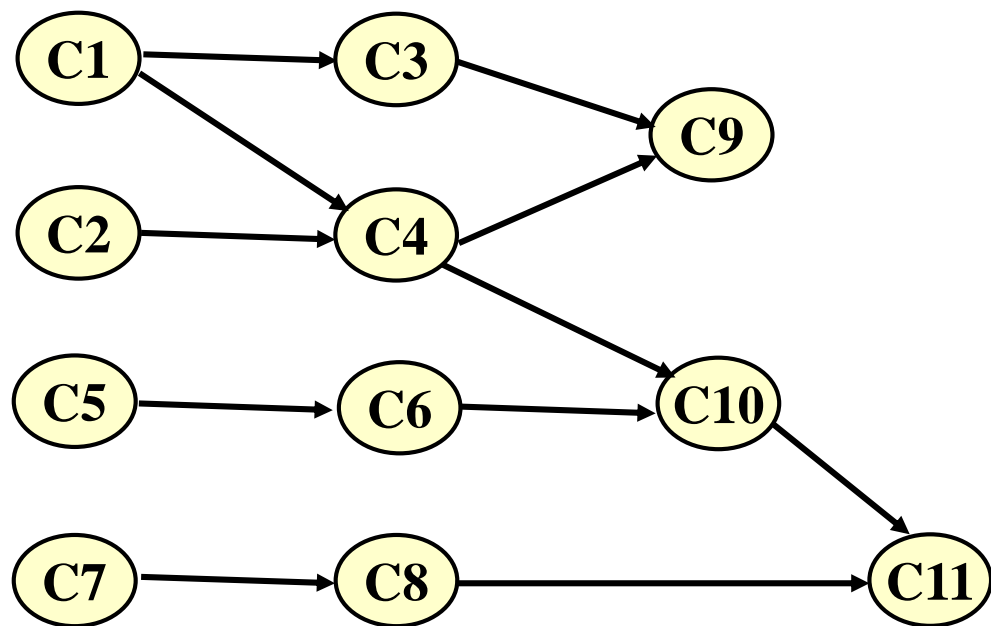
AOV网：顶点活动网(Activity On Vertex network)



课程编号	课程名称	先修课程
C1	高级程序设计	无
C2	离散数学	无
C3	计算机组成原理	C1
C4	数据结构	C1,C2
C5	高等数学	无
C6	信息论	C5
C7	计算机导论	无
C8	计算机网络	C7
C9	操作系统	C3,C4
C10	密码学	C4,C6
C11	网络安全技术	C10,C8

拓扑排序方法

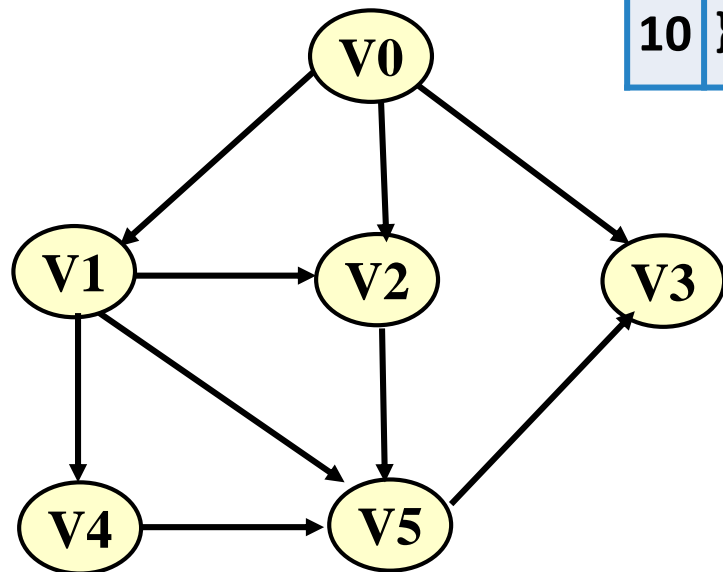
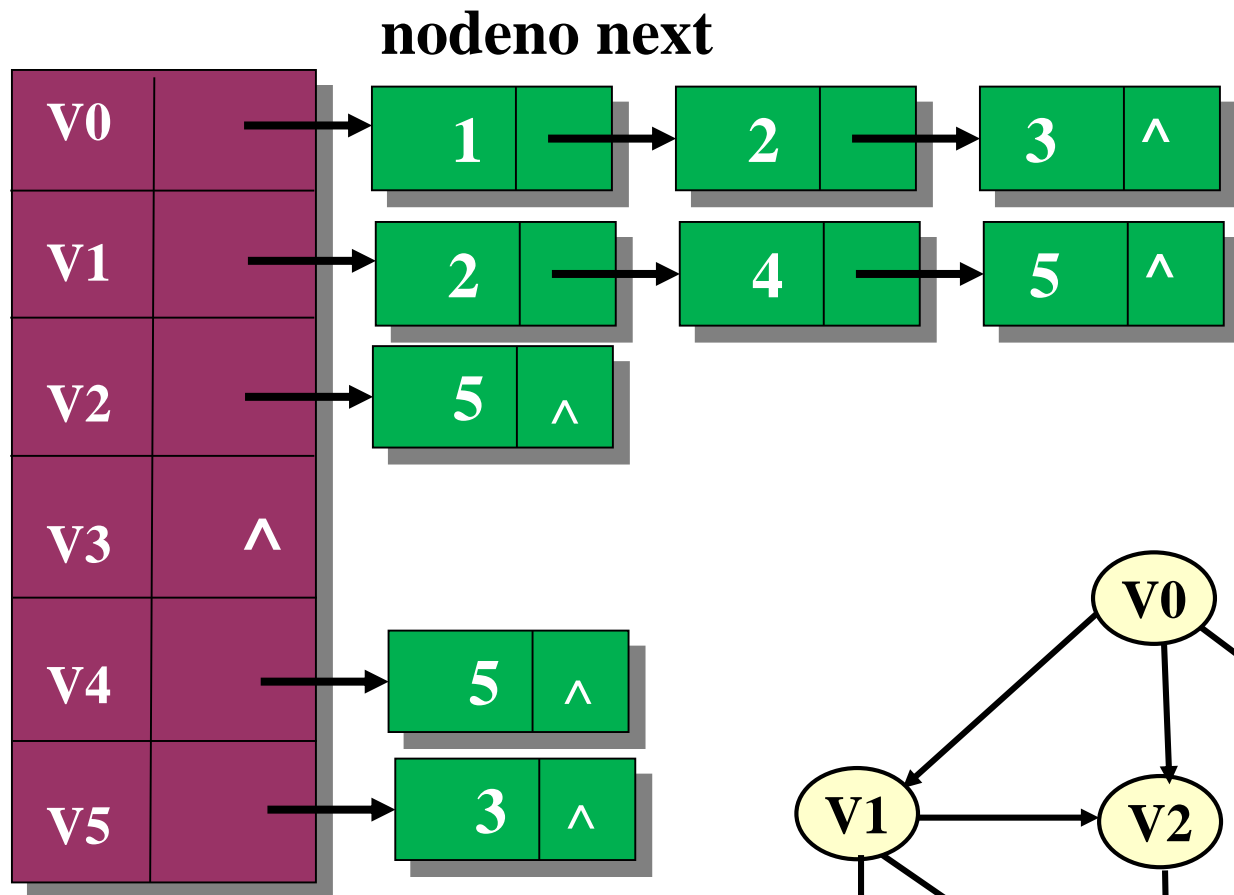
- ◆ 输出AOV中一个没有前驱结点的顶点；
- ◆ 在AOV中删除该顶点以及对应的出边
- ◆ 重复上述两步，直到所有顶点都输出为止，完成了拓扑排序；或者还有未输出的顶点，这些顶点有前驱而不能删除，说明存在环路，这样的工程是不可行的。



一个AOV网的拓扑序列不是唯一的

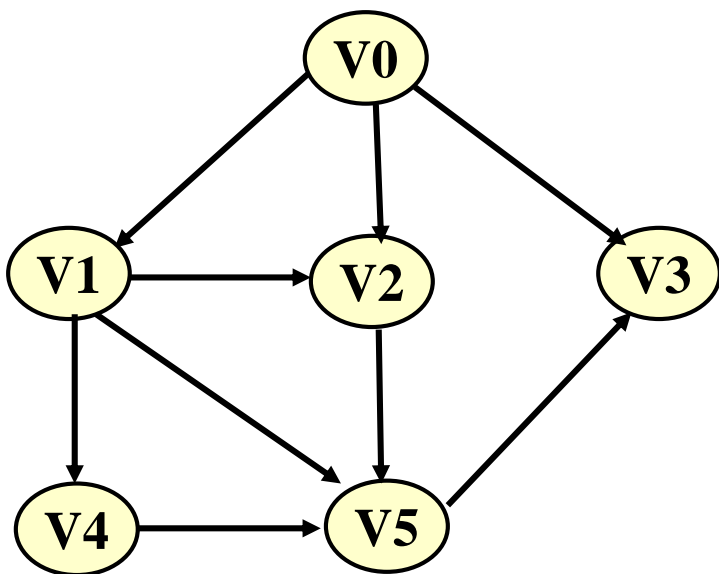
AOV网中如果出现回路一定不能完成拓扑排序

6.7 拓扑排序



```
1 typedef struct    GRAPHLISTNODE_STRU
2 {
3     int nodeno;    //图中结点的编号
4     struct    GRAPHLISTNODE_STRU* next;
5 }GraphListNode;
6 typedef struct    GRAPHLIST_STRU
7 {
8     int size;      //图中实际的结点个数
9     GraphListNode* graphListArray;
10    //图的顶点表，用二维数组表示
11 }GraphList;
```

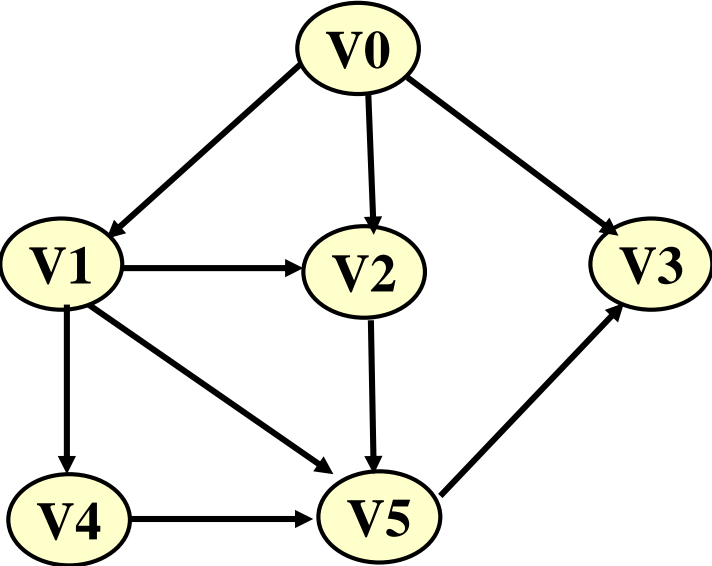
6.7 拓扑排序



拓扑排序算法核心过程：

1. 计算各个顶点的入度；
2. 将入度为0的顶点入栈；
3. 如果栈不空，从栈中取出一个元素 v ，输出到拓扑序列中；
4. 检查顶点 v 的出边表，将出边表中的每个顶点 w 的入度减1（即删除顶点 v 为弧头的边表），如果 w 的入度为0，则顶点 w 入栈
5. 重复第三步和第四步，直到栈为空结束

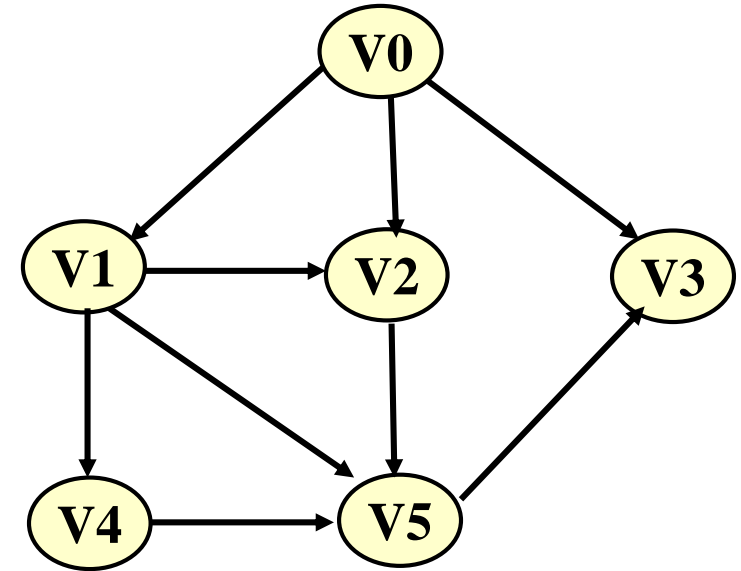
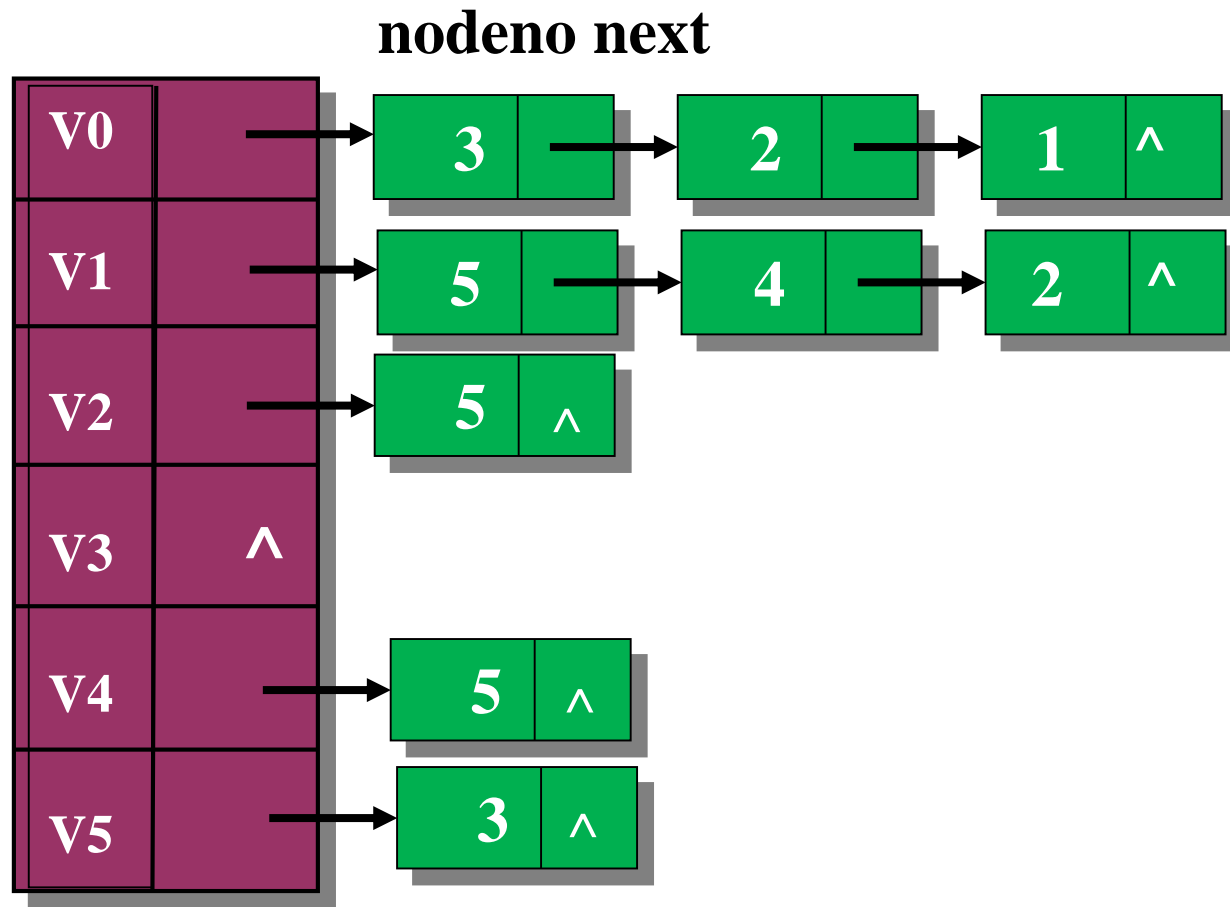
6.7 拓扑排序



蓝色表示该顶点已经输出到拓扑序列
红色表示顶点的入度有更新
绿色表示栈中内容

拓扑序列	各顶点的入度inPoint[]						栈nodeStack 内容
	V0	V1	V2	V3	V4	V5	
初始	0	1	2	2	1	3	V0
V0	0	0	1	1	1	3	V1
V0 V1	0	0	0	1	0	2	V2 V4
V0 V1 V2	0	0	0	1	0	1	V4
V0 V1 V2 V4	0	0	0	1	0	0	V5
V0 V1 V2 V4 V5	0	0	0	0	0	0	V3
V0 V1 V2 V4 V5 V3	0	0	0	0	0	0	

6.7 拓扑排序



6.7 拓扑排序

