7.5.1 拓扑排序

- 无环的有向图称为有向无环图，简称DAG图(Directed Acyclic Graph)，它可以表示一个工程或系统的流程图。
  - 一项大工程可以分为若干个称为活动的子工程，用顶点表示；某些子工程必须在另一些子工程完成之后才能开始，用弧表示它们之间的前趋后继关系；这样构成的有向图显然是无环的。
  - 如何使工程顺利进行？
  - 完成总工程需要的最短时间？
- 就归结为拓扑排序和关键路径问题。

7.5.1 拓扑排序

- 假设以有向图表示一个工程的施工图或程序的数据流图，每个顶点代表一个活动，弧<v_i，v_j>表示活动 *i* 必须先于活动 *j* 进行，称为**AOV-网**(Activity On Vertex)，图中不允许出现回路。

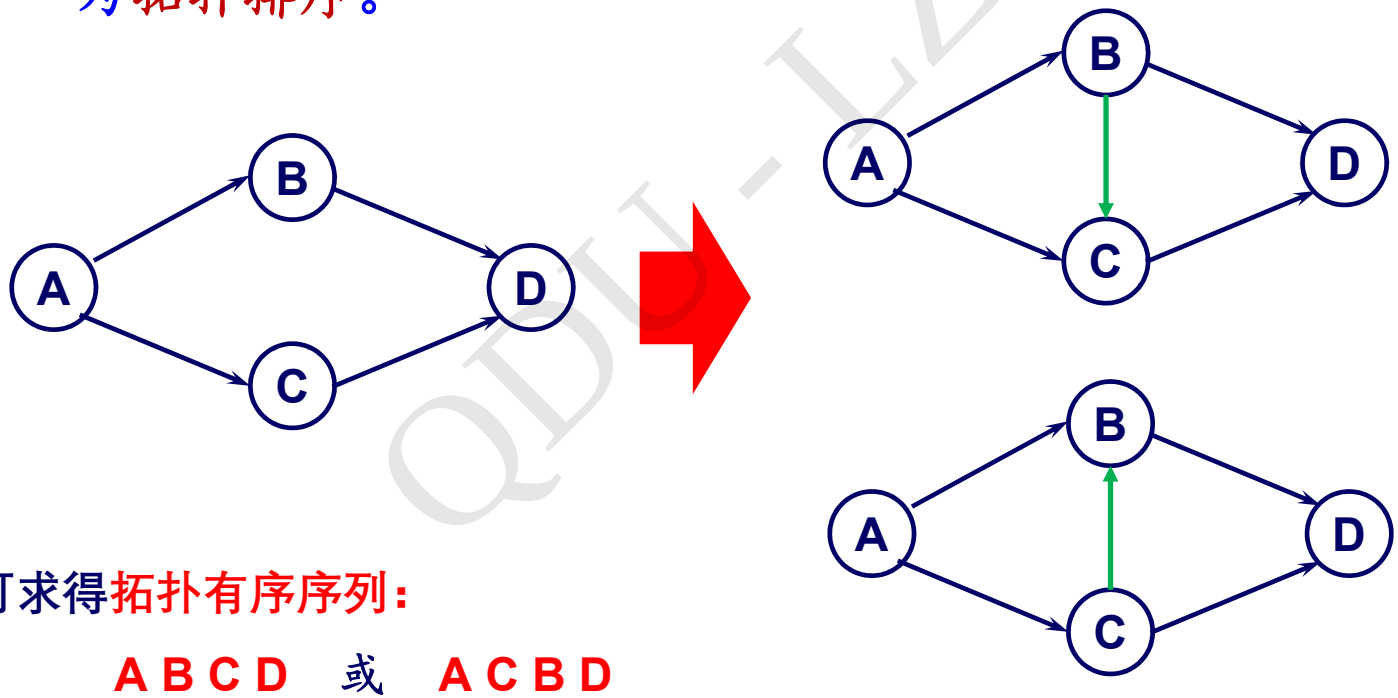
➢ 检查有向图中是否存在回路的方法之一，是对有向图进行**拓扑排序**。

7.5.1 拓扑排序

- AOV网中顶点序列v_1、v_2、…、v_n称为一个拓扑序列，当且仅当该顶点序列满足下列条件：若<v_i,v_j>是图中的弧，则在序列中顶点v_i必须排在顶点v_j之前。
- 在一个有向图中，将图中顶点排成一个线性序列，对于有向图中没有限定次序关系的顶点，则可以人为加上任意的次序关系。
- 找一个拓扑序列的过程称为拓扑排序。

7.5.1 拓扑排序

- 拓扑排序（Topological Sort）：由某个集合上的一个偏序得到该集合上的一个全序，这个操作称之为拓扑排序。

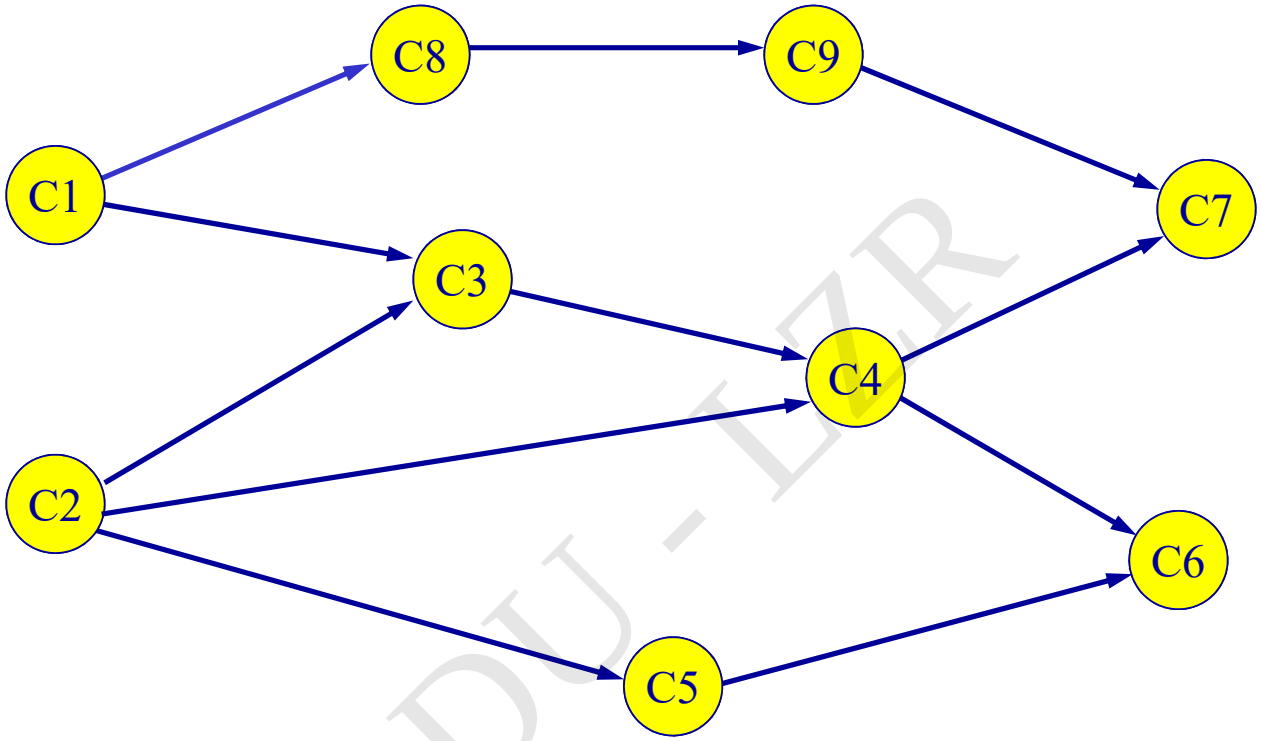


可求得拓扑有序序列：

A B C D　或　A C B D

例如，计算机专业的学生必须完成一系列规定的基础课和专业课才能毕业，假设这些课程的名称与相应编号如下表所示。

课程代号	课程名称	先修课程
C_1	高等数学	
C_2	程序设计基础	
C_3	离散数学	C_1, C_2
C_4	数据结构	C_3, C_2
C_5	高级语言程序设计	C_2
C_6	编译方法	C_5, C_4
C_7	操作系统	C_4, C_9
C_8	普通物理	C_1
C_9	计算机原理	C_8



学生课程学习工程图

可求得拓扑有序序列：C1 C2 C8 C5 C3 C9 C4 C7 C6

拓扑排序步骤如下：

（1）从AOV网中选择一个没有前驱（即入度为0）的顶点并且输出它。

（2）从AOV网中删去该顶点，并且删去从该顶点发出的全部有向边。

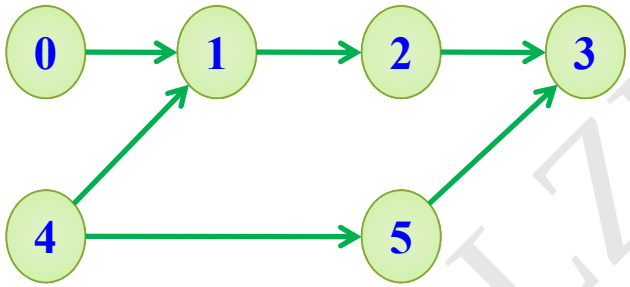
（3）重复上述两步，直到剩余的网中不再存在没有前驱的顶点为止。

对任一有向图进行拓扑排序有两种结果：

- 图中全部顶点都包含在拓扑序列中，这说明该图中不存在有向回路；
- 图中部分顶点未被包含在拓扑序列中，这说明该图中存在有向回路。

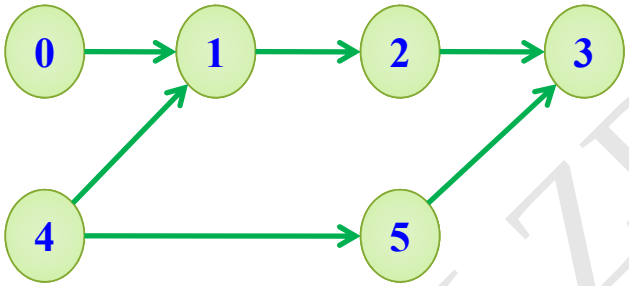
- 所以可以采用拓扑排序判断一个有向图中是否存在回路。

【示例-1】 给出下图G的全部可能的拓扑排序序列。



解：从图中看到，入度为0有两个顶点，即0和4。

先考虑顶点0：删除0及相关边，入度为0者有4；删除4及相关边，入度为0者有1和5；考虑顶点1，删除1及相关边，入度为0者有2和5；如此得到拓扑序列：041253，041523，045123。



再考察顶点4，类似地得到拓扑序列：450123，401253，405123，401523。

因此，所有的拓扑序列为：041253，041523，045123，450123，401253，405123，401523。

拓扑排序算法设计

将邻接表定义中的VNode类型修改如下：

```
typedef struct              //表头结点类型
{    Vertex data;           //顶点信息
     int count;             //存放顶点入度
    ArcNode *firstarc;      //指向第一条边
} VNode;
```

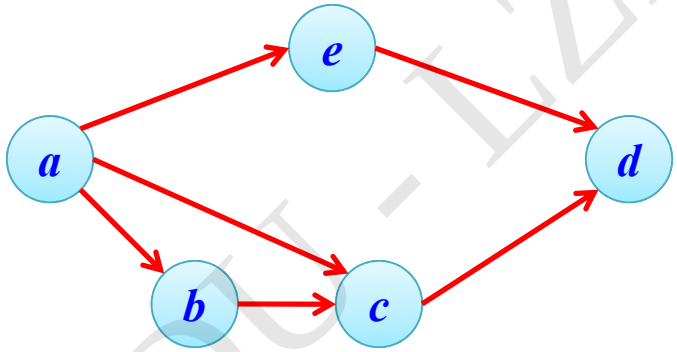
拓扑排序算法如下：

```
void TopSort(ALGraph *G)          //拓扑排序算法
{
    int i，j;
    int S[MAXV]，top=-1;          //栈S的指针为top
    ArcNode *p;
    for (i=0;i<G->n;i++)          //入度置初值0
        G->adjlist[i].count=0;
    for (i=0;i<G->n;i++)          //求所有顶点的入度
    {   p=G->adjlist[i].firstarc;
        while (p!=NULL)
        {     G->adjlist[p->adjvex].count++;
              p=p->nextarc;
        }
    }
```

```
    for (i=0;i<G->n;i++)                    //将入度为0的顶点进栈
        if (G->adjlist[i].count==0)
            S[++top]=i;
    while (top>-1)                          //栈不空循环
    {
        i=S[top--];                         //出栈一个顶点i
        printf("%d ", i);                   //输出该顶点
        p=G->adjlist[i].firstarc;           //找第一个邻接点
        while (p!=NULL)                     //将顶点i的出边邻接点的入度减1
        {   j=p->adjvex;
            G->adjlist[j].count--;
            if (G->adjlist[j].count==0)     //将入度为0的邻接点进栈
                S[++top]=j;
            p=p->nextarc;                   //找下一个邻接点
        }
    }
}
```

【示例-2】对如图所示的图进行拓扑排序，可以得到不同的拓扑序列个数是_____。

A. 4　　B. 3　　C. 2　　D. 1



注：2010年全国考研题

解：不同的拓扑序列有：*aebcd*、*abced*、*abecd*。答案为B。

END