

有向图的拓扑排序算法

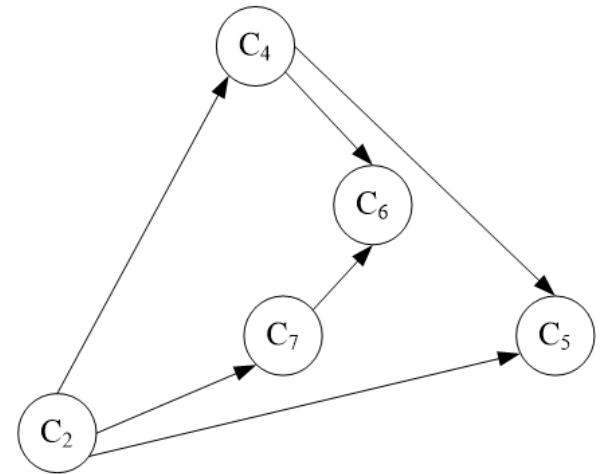
Outline

- 有向无环图与AOV网的概念
- 拓扑排序的概念
- 拓扑排序的算法设计与实现方法
- 拓扑排序算法的用途

有向无环图的概念

- 有向无环图 (Directed Acyclic Graph, DAG)

- 指一个不含环 (回路) 的有向图
- 注意区分下列概念的区别
 - 有向树
 - 有向无环图
 - 有向图

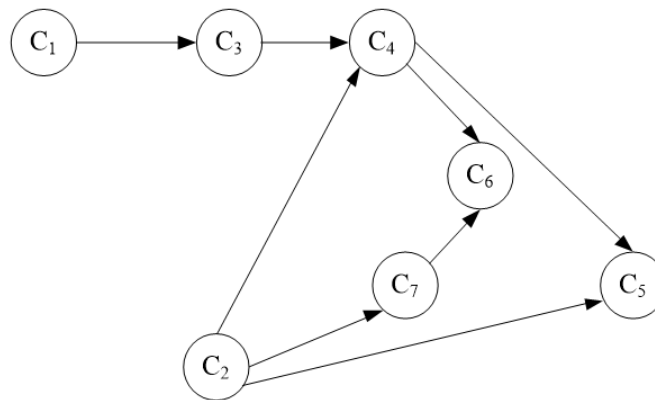


- 有向无环图可以以一种自然的方式用于对**优先关系**或**依赖关系**进行描述，因而在工程计划与管理方面有着广泛而重要的应用。

AOV网的概念

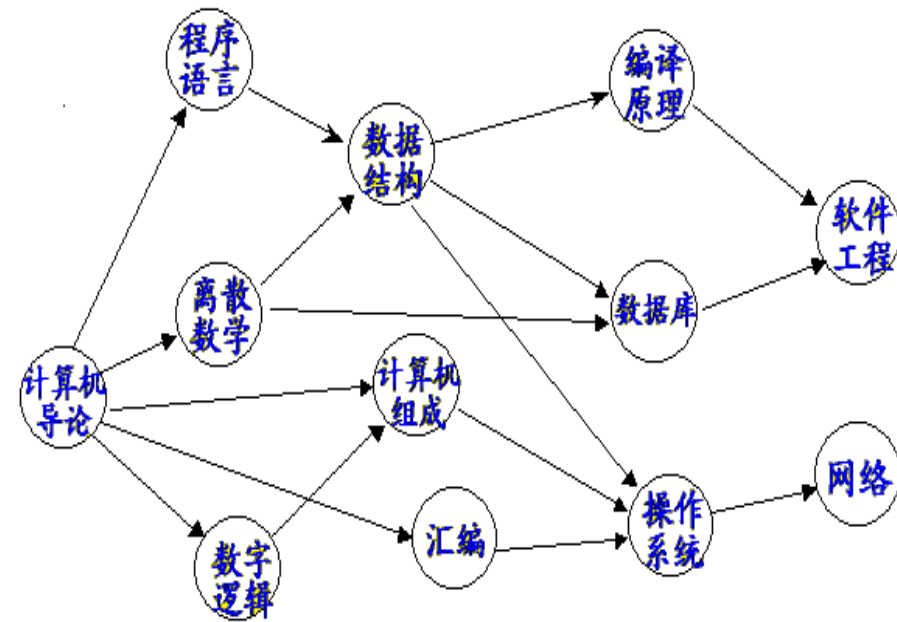
- AOV网 (Activity on Vertex Network)

- 一个大的工程往往可以分解为若干具有相对独立性的**子工程（活动）**，子工程之间在进行的时间上有一定的相互制约关系，将这些子工程之间的先后关系用有向图表示，其中顶点表示**活动**，有向边表示活动之间的**优先制约关系**，称这种有向图为顶点表示活动的网，简称**AOV网**。



AOV网的举例

- 顶点--- 表示活动
- 有向边--- 表示活动间的优先关系
- 前驱--- 若从 V_i 到 V_j 有路径，则称 V_i 是 V_j 的前驱。
- 后继--- 若从 V_i 到 V_j 有路径，则称 V_j 是 V_i 的后继。



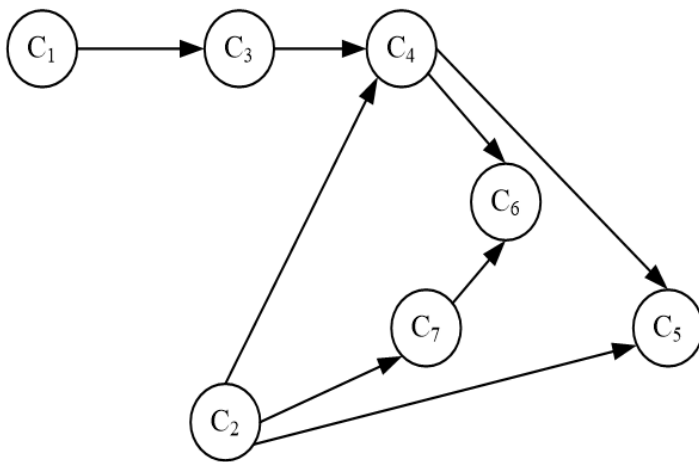


拓扑排序的概念

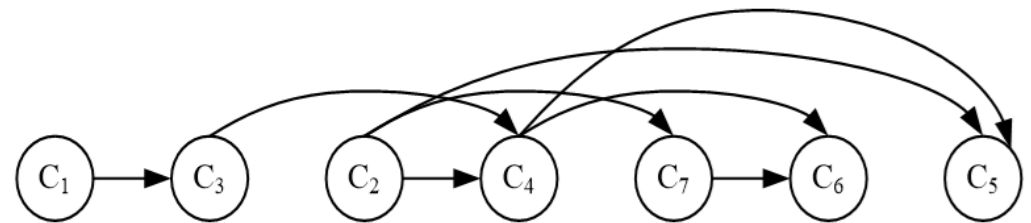
- 对一项工程采用AOV网表示以后，为判断该工程的可行性，需要检查AOV网中是否存在环。
 - 因为环的出现意味着某项活动的开工将以自身工作的完成为先决条件，这种情况称为**死锁**。
- 检测有向图中是否存在环的一种常用方法就是对一个有向图进行**拓扑排序**(Topological Sort)。

拓扑排序的概念

- 拓扑排序：将AOV网中的各个顶点 (代表各个活动) 排列成一个线性有序的序列 (**拓扑序列**)，使得AOV网络中所有应存在的前驱和后继关系都能得到满足。
- 由某个集合上的一个偏序得到该集合上的一个全序。



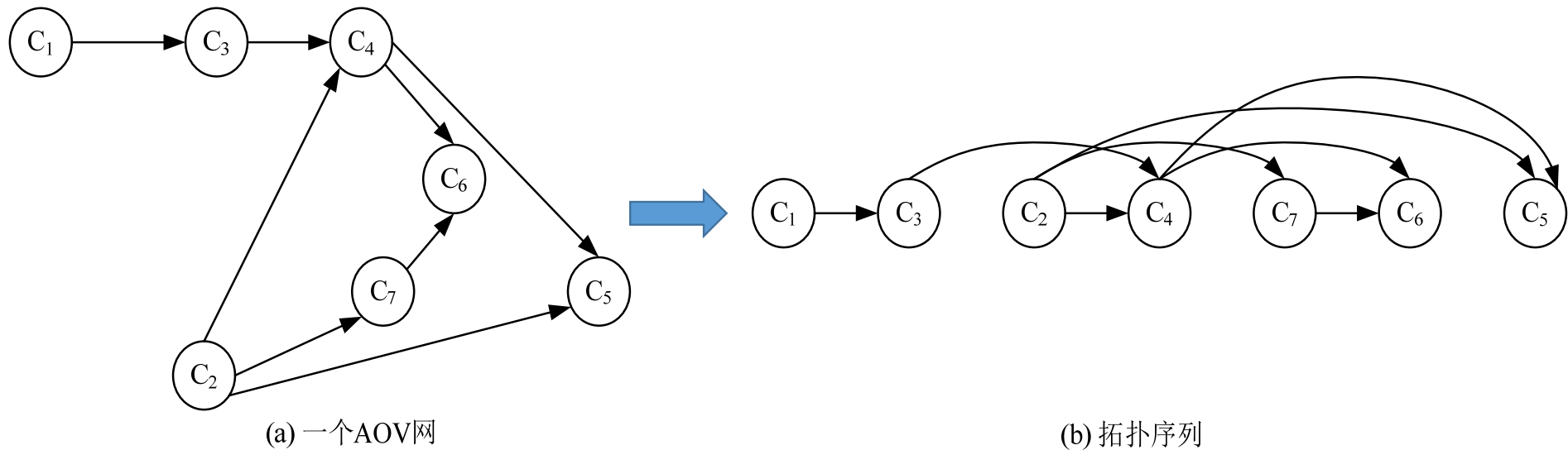
(a) 一个AOV网



(b) 拓扑序列

拓扑排序的实现方法

- 如何实现将AOV网中的各个顶点线性化？
 - 通过对图深度优先搜索DFS的扩展实现拓扑排序
 - 基于顶点入度动态计算的拓扑排序方法

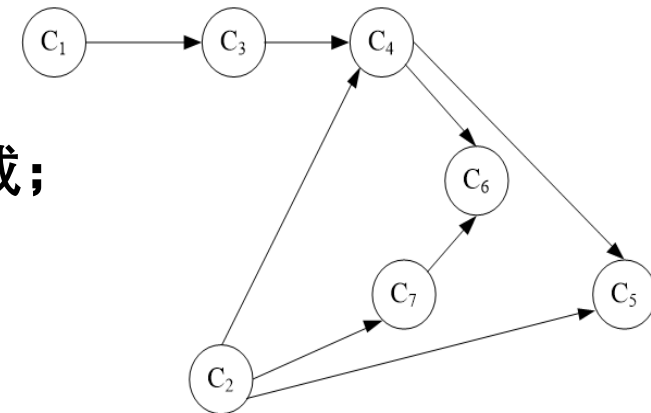


基于顶点入度动态计算的拓扑排序法

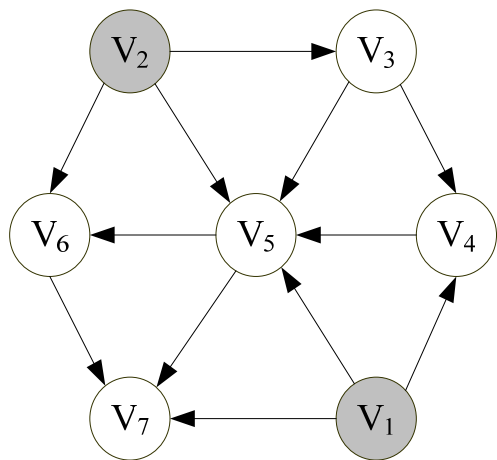
- 1、在AOV网中选一个没有直接前驱的顶点（即此顶点入度为0），并输出之；
- 2、从图中删去该顶点，同时删去所有从它发出的有向边；

◆ 重复以上两步，直到：

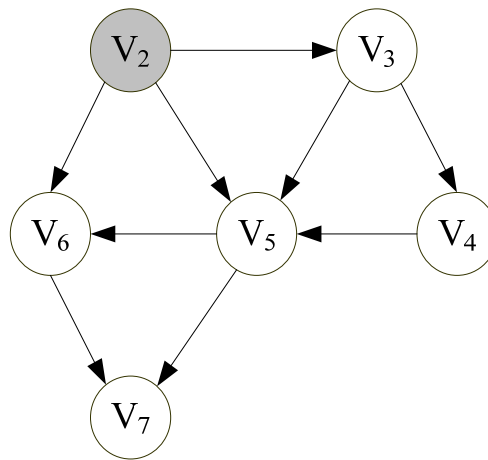
- ✓ 全部顶点均已输出，拓扑有序序列形成，拓扑排序完成；
- ✓ 图中还有未输出的顶点，但已跳出处理循环。
 - 这个AOV网络中必定存在有向环



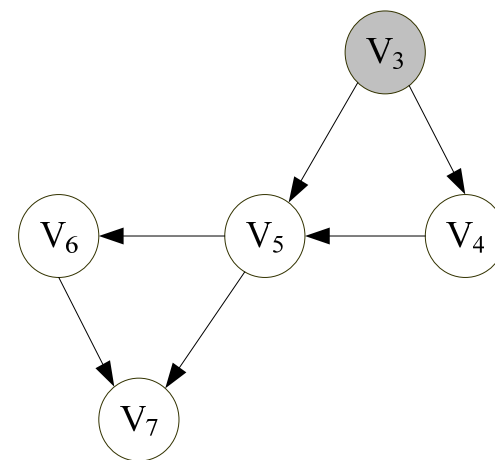
拓扑排序过程示例



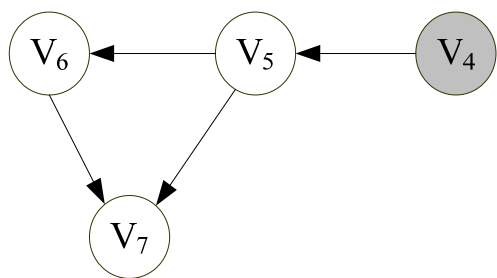
(a) AOV网



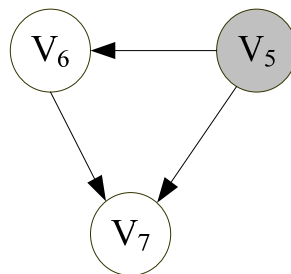
(b)



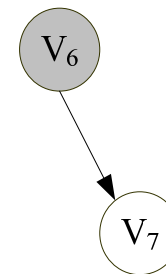
(c)



(d)



(e)



(f)

输出的顶点拓扑序列: $V_1 - V_2 - V_3 - V_4 - V_5 - V_6 - V_7$



拓扑排序算法的实现方法

- 实现上述拓扑排序过程，需要考虑以下三个问题：

- (1) 图的存储结构怎么选择？

- (2) 在每次选出一个入度为0的顶点后，如何实现“从图中删去该顶点，同时删去所有从它发出的有向边”的操作？

- (3) 在拓扑排序过程中，如何方便地依次选出各个入度为0的顶点？

拓扑排序算法的实现方法

- 问题(1): 图的存储结构?
 - ✓ 选择邻接表作为存储结构更合适
- 问题(2): 如何实现“从图中删去该顶点,同时删去所有从它发出的有向边”的操作?
 - ✓ 可以用“弧头顶点的入度减1”的方法来间接实现。
 - ✓ 为便于考察每个顶点的入度,引入一个存放各顶点入度的数组 (`indegree[]`)
- 问题(3): 如何方便地依次选出各个入度为0的顶点?
 - ✓ 为了避免每一步选入度为0的顶点时,重复扫描数组indegree,设置一个队列(或栈)来存储所有入度为0的顶点。
 - ✓ 在进行拓扑排序之前,只要对顶点表扫描一遍,将所有入度为0的顶点都入队,一旦排序过程中出现新的入度为0的顶点,也同样将其入队。



拓扑排序的算法描述框架

1. 扫描顶点表，将入度为0 的顶点入队；
2. while (队列非空) {
 将队头顶点 v 删除并输出之；
 检查 v 的出边，将每条出边 $\langle v, u \rangle$ 终点 u 的入度减1，
 若 u 的入度变为0，则将 u 入队；
}
3. 若输出的顶点数小于 n ，则输出“有回路”； 否则拓扑排序正常结束。

```

template<class T>
void ALGraph<T>::TopoSort(){
    int *indegree=new int[vexnum];
    SeqQueue<int,20> s;
    for(i=0;i<vexnum;i++) indegree[i]=0;
    EdgeNode *p;
    for(i=0;i<vexnum;i++) //求所有顶点的入度
        for(p=adjlist[i].firstedge; p; p=p->nextedge) indegree[p->adjvex]++;
    for(i=0;i<vexnum;i++) //使入度为0的顶点入队
        if(!indegree[i]) s.Enqueue(i);
    c=0; //对输出顶点计数
    while(!s.Empty()){
        i=s.DeQueue(); ++c; //出队
        cout<<adjlist[i].data; //输出顶点
        for(p=adjlist[i].firstedge; p; p=p->nextedge){
            indegree[p->adjvex]--;
            if(!indegree[p->adjvex]) s.Enqueue(p->adjvex);
        }
    }
    if(c<vexnum) cout<<"该 AOV 网存在回路!"<<endl;
    delete[] indegree;
}

```

算法的时间复杂度
为 $O(n+e)$

拓扑排序算法的用途

• 用途？

- ✓ 将有向无环图中的顶点线性化；
 - 判断一个大的工程的可行性
 - 可应用于一些基于图的动态规划算法设计过程
- ✓ 检测一个有向图中是否存在环。

思考题



1. 如何采用图的深度优先搜索过程实现拓扑排序？
2. 采用拓扑排序算法可以判别一个有向图中是否存在有向环。
除此之外，你还可以设计什么算法判别有向环的存在？

Thank you for your attention!