

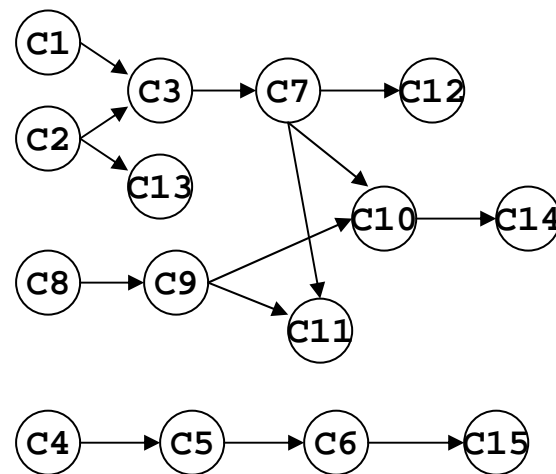
第八讲 图（下）

浙江大学 陈 越

8.2 拓扑排序

例：计算机专业排课

课程号	课程名称	预修课程
C1	程序设计基础	无
C2	离散数学	无
C3	数据结构	C1, C2
C4	微积分（一）	无
C5	微积分（二）	C4
C6	线性代数	C5
C7	算法分析与设计	C3
C8	逻辑与计算机设计基础	无
C9	计算机组成	C8
C10	操作系统	C7, C9
C11	编译原理	C7, C9
C12	数据库	C7
C13	计算理论	C2
C14	计算机网络	C10
C15	数值分析	C6



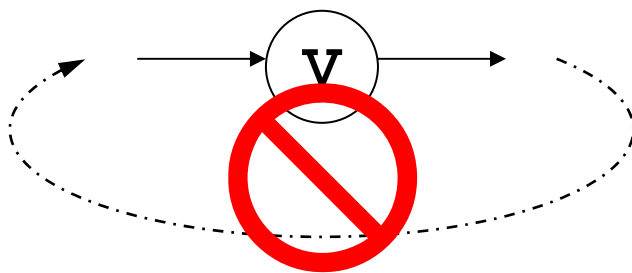
AOV (Activity On Vertex)

网络



拓扑排序

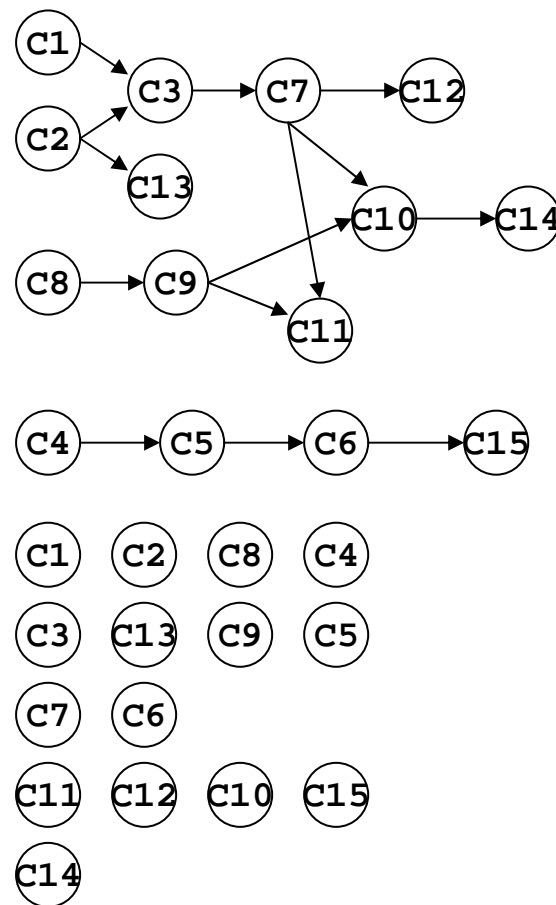
- **拓扑序**：如果图中从 v 到 w 有一条有向路径，
☞ 则 v 一定排在 w 之前。满足此条件的顶点序列称为一个拓扑序
- 获得一个拓扑序的过程就是**拓扑排序**
- **AOV**如果有**合理的**拓扑序，则必定是**有向无环图**（**Directed Acyclic Graph, DAG**）



v 必须在 v 开始之前结束


算法

课程号	课程名称	预修课程
C1	程序设计基础	无
C2	离散数学	无
C3	数据结构	C1, C2
C4	微积分（一）	无
C5	微积分（二）	C4
C6	线性代数	C5
C7	算法分析与设计	C3
C8	逻辑与计算机设计基础	无
C9	计算机组成	C8
C10	操作系统	C7, C9
C11	编译原理	C7, C9
C12	数据库	C7
C13	计算理论	C2
C14	计算机网络	C10
C15	数值分析	C6



算法

```
void TopSort()  
{  for ( cnt = 0; cnt < |V|; cnt++ ) {  
    v = 未输出的入度为0的顶点; /* O(|V|) */  
    if ( 这样的v不存在 ) {  
        Error ( "图中有回路" );  
        break;  
    }  
    输出v, 或者记录v的输出序号;  
    for ( v 的每个邻接点 w )  
        Indegree[W]--;  
  }  
}
```

 $T = O(|V|^2)$

聪明的算法

- 随时将入度变为**0**的顶点放到一个容器里

```
void TopSort()  
{ for ( 图中每个顶点 v )  
    if ( Indegree[V]==0 )  
        Enqueue( V, Q );  
    while ( !IsEmpty(Q) ) {  
        V = Dequeue( Q );  
        输出v, 或者记录v的输出序号; cnt++;  
        for ( v 的每个邻接点 w )  
            if ( --Indegree[W]==0 )  
                Enqueue( W, Q );  
    }  
    if ( cnt != |V| )  
        Error( "图中有回路" );  
}
```

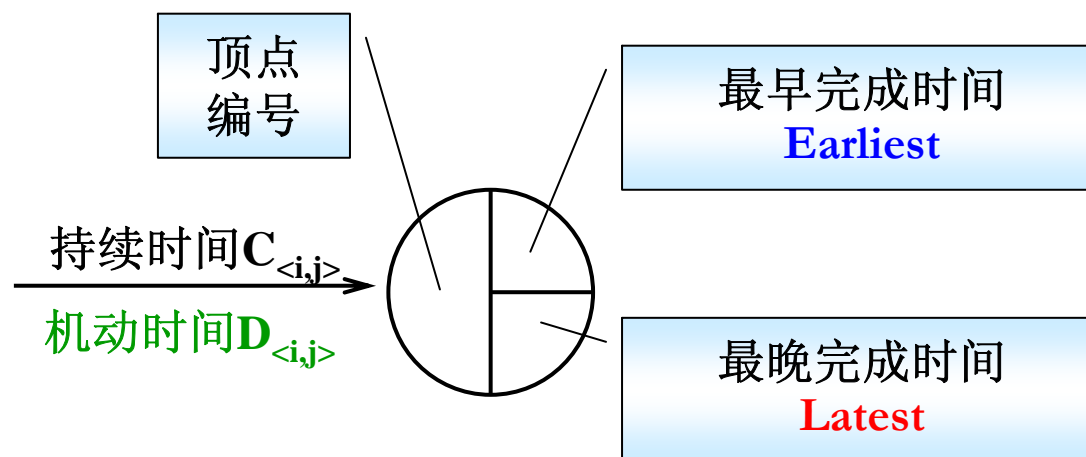
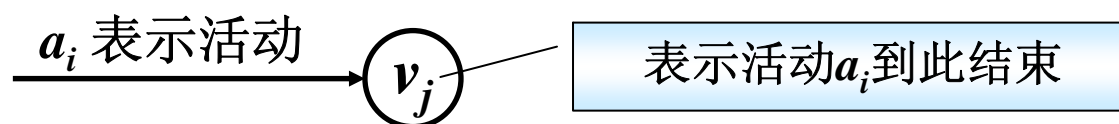
👍 $T = O(|V| + |E|)$

此算法可以用来
检测有向图是否
DAG

关键路径问题

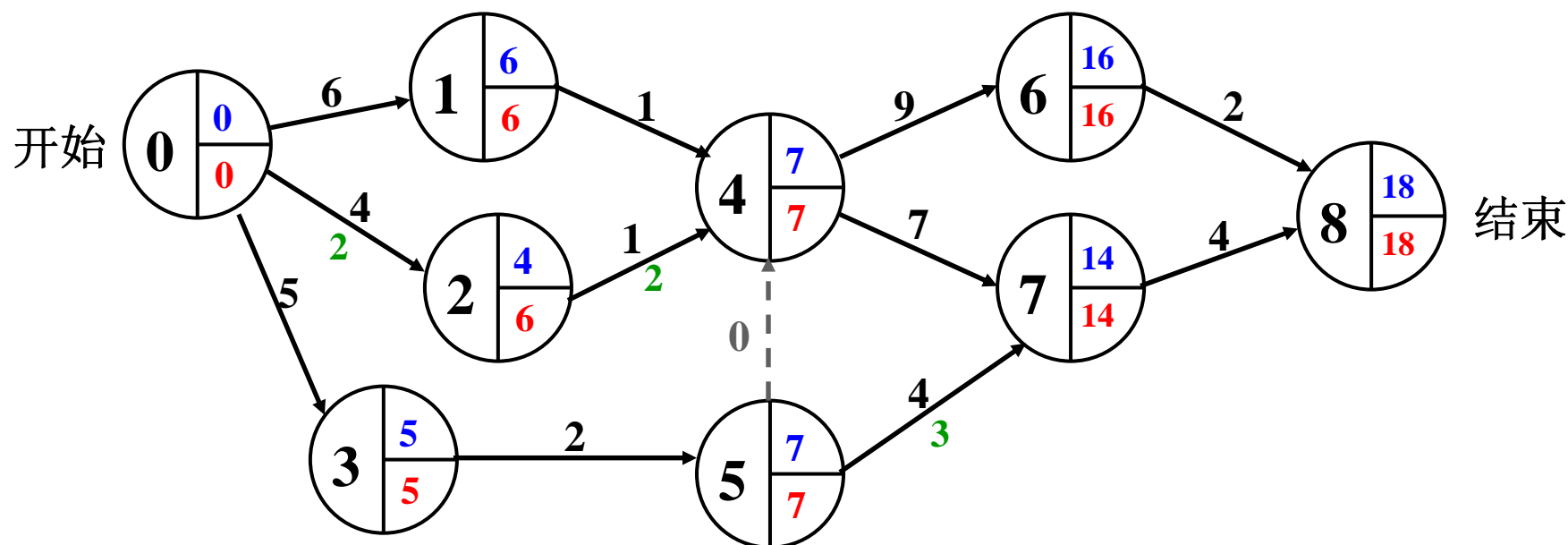
■ AOE (Activity On Edge) 网络

- 一般用于安排项目的工序



关键路径问题

由绝对不允许延误的活动组成的路径



问题1: 整个工期有多长? $\text{Earliest}[8] = 18$

$\text{Earliest}[0] = 0;$

$\text{Earliest}[j] = \max_{\langle i,j \rangle \in E} \{ \text{Earliest}[i] + C_{\langle i,j \rangle} \};$

问题2: 哪几个组有机动时间? $D_{\langle i,j \rangle} = \text{Latest}[j] - \text{Earliest}[i] - C_{\langle i,j \rangle}$

$\text{Latest}[8] = 18;$

$\text{Latest}[i] = \min_{\langle i,j \rangle \in E} \{ \text{Latest}[j] - C_{\langle i,j \rangle} \};$