

拓扑排序 关键路径

2019年7月1日 15:26

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)

■ 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" );
}
```

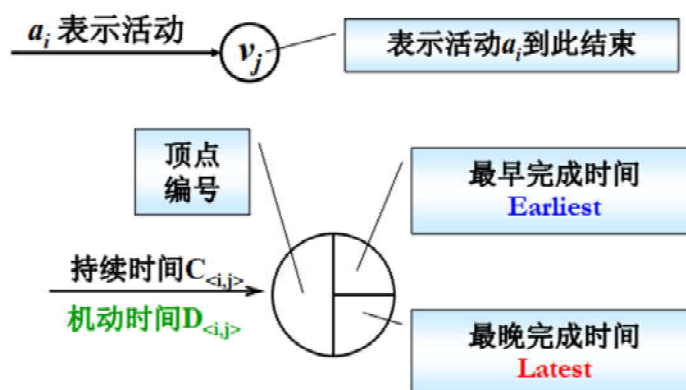
👍 $T = O(|V| + |E|)$

此算法可以用来检测有向图是否 DAG

关键路径问题

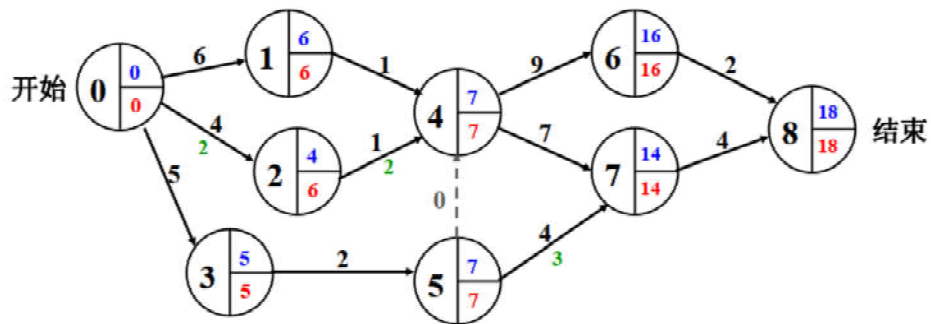
■ AOE (Activity On Edge) 网络

- 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? $D_{\langle i, j \rangle} = \text{Latest}[j] - \text{Earliest}[i] - C_{\langle i, j \rangle}$

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

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