

实验八 拓扑排序

【问题描述】

建造一座办公楼房，需要进行选择设计单位、楼房总体设计、选择施工企业等活动。活动之间存在时序上的制约关系。例如，活动 a 在 b 完成之后开始，则称 b 为 a 的紧前活动。下表是一个活动清单，体现活动之间的关系。请根据此清单画出该建造办公楼项目的 AOV 网。

活动代号	活动名称	紧前活动
A1	选择设计单位	无
A2	楼房总体设计	A1, A4, A5
A3	选择施工企业	A2
A4	选择建造地点	无
A5	资金筹措	无
A6	购买建筑材料	A3, A5
A7	建造地基	A6
A8	建造楼房	A7
A9	楼房封顶	A8
A10	购买装修材料	A5
A11	外部装修	A9, A10
A12	内部装修	A11
A13	工程验收	A11, A12

【实验要求】

请设计程序判断该项目是否有回路，若有则提示有回路，若无请输出该项目 AOV 网的拓扑序列。

用户键盘输入 AOV 网的各项点偶对（如输入 3,4 表示顶点 3 到顶点 4 有一条有向边；输入 0,0 表示结束输入），程序建立有向图的邻接表，并进行拓扑排序，输出结果。

【算法提示】

课堂上讲过拓扑排序的方法。我们可采用邻接表作有向图的存储结构，且在头结点中增加一个存放顶点入度的数据域。入度为 0 的顶点即为没有前驱的结点，可查找表头数组求得；至于删除顶点及以它为尾的弧的操作，则可换以弧头顶点的入度减 1 来实现。

例如，图 1 中有向图的邻接表如图 2 所示，其中 V1 和 V6 的入度为 0，在输出 V6 之后可将 V5 和 V4 的入度分别减为 2 和 1。为了避免重复检测入度为 0 的顶点，可另设一链表将所有入度为 0 的顶点链接在一起，每当输出便从链表中删除。反之，当出现新的入度为 0 的顶点则可插入。同时，由于入度为 0 的顶点在拓扑有序序列中的次序是人为设定的，因此，插入和删除操作均可在表头进行，此时的链表相当于一个链栈。进行拓扑排序后，若栈空时输出的顶点数 m 不足有向图中的顶点数 n，则说明有向图中存在有向环，否则拓扑排序完成。

可编写两个子函数：建立邻接表、拓扑排序。

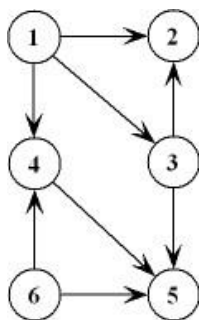


图 1 有向图

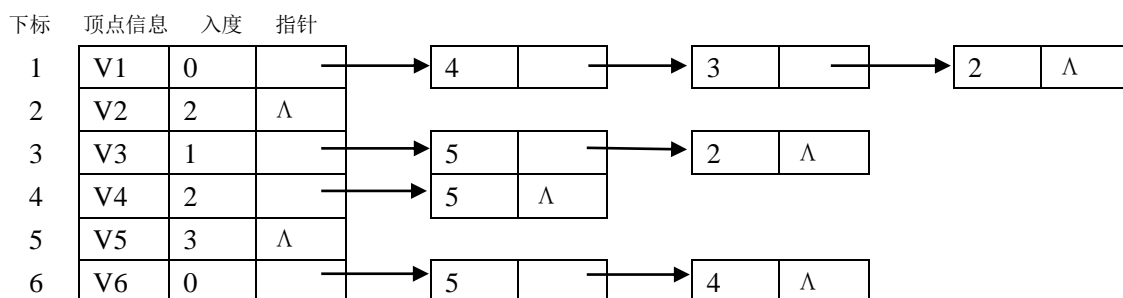


图 2 图 1 中有向图的邻接表

在实现以上思路的基础上，要求进一步利用一个技巧节省为链栈另分配的存储空间。

对应于图 2 的邻接表，链栈的初始状态为图 3，它表明此时有两个入度为 0 的顶点。可以借用值为 0 的入度域存放链栈中的指针（以指示下一个入度为 0 的顶点），而链栈中的顶点号即为该顶点在邻接表中的序号，由此，和上述链栈相应的入度域的状态如图 4(b)所示。图 4(c)指示在输出 v6 之后入度为 0 的顶点为 v1；而图 4(d)表明在输出 v1 之后，由于 v4 和 v3 的入度相继为 0，则栈顶指针指向 v3，而 v3 指向 v4。依次类推。

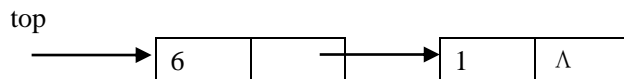


图 3

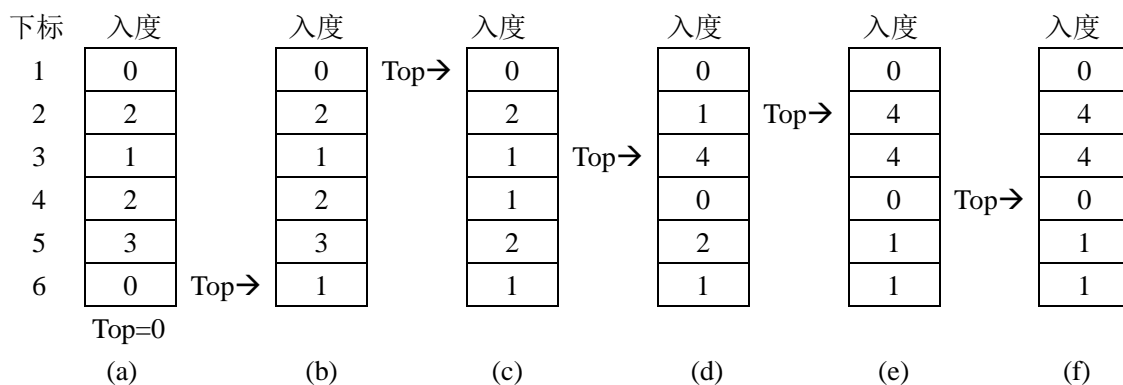


图 4 拓扑排序过程中入度域的变化过程例示

(a) 邻接表建成后的状态；(b) 进行拓扑排序之前的状态；(c) v6 出栈之后的状态；
(d) 输出 v1 之后的状态；(e) 输出 v3 之后的状态；(f) v2 出栈之后的状态。