

该算法是文本中介绍的基本拓扑排序算法的一个推广，其核心思想是利用拓扑排序的迭代过程来检测有向图中的环。

1. **拓扑排序阶段**：在每次迭代中，我们首先尝试寻找图中**入度为零**的结点。

- 如果找到这样的结点，则将其从图中移除（概念上），并将其指向的结点的入度减一。这个过程会不断迭代，直到所有入度为零的结点都被处理。如果最终能处理完所有结点，则图是**有向无环图 (DAG)**，并且我们得到了一个拓扑排序。

2. **环检测阶段**：如果在某次迭代中，**所有剩余结点都至少有一个入度**，这意味着无法找到入度为零的结点。根据拓扑排序理论，这种情况发生当且仅当图  $G$  中包含一个**有向环**。

3. **环的查找 ( $O(n)$ )**：一旦确定存在环，我们可以利用每个结点至少有一个入度的特性，以  $O(n)$  的运行时间实际找到一个环。具体步骤如下：

- **起始**：随意选择一个结点作为当前结点。
- **重复跟随入边**：从当前结点开始，重复沿着它的**任意一条入边**（例如，选择其入边邻接列表中的第一条）回溯到前一个结点。
- **终止**：由于图中每个结点都有入度，这个回溯过程可以无限进行下去，直到我们第一次**重新访问**到一个已经遇到过的结点  $v$ 。
- **识别环**：从第一次访问  $v$  到第二次访问  $v$  之间所遇到的所有结点（不包括重复的  $v$ ，或者说闭合路径）构成了一个有向环  $C$ 。这个环是沿着回溯路径（即原图中的逆方向）形成的。

## 运行时间

- **拓扑排序阶段**：基于入度的拓扑排序算法通常可以在  $O(n + m)$  的时间内完成，其中  $n$  是结点数， $m$  是边数。
- **环查找阶段**：一旦确定存在环，我们通过重复跟随入边来查找环。由于每个结点只会被重新访问一次，并且每次操作都是常数时间（选择第一个入边），这个查找过程的运行时间是  $O(n)$ （因为最多回溯  $n$  个结点）。

因此，整个算法的运行时间复杂度取决于拓扑排序阶段，即  $O(n + m)$ 。