拓扑排序的定义：包含n个节点的DAG，给出一个节点编号的排列，如果满足对G的任意一条有向边(u,v)，u在排列中都出现在v的前面，就称为是G的拓扑排序。

结论：1.如果存在环则不存在拓扑排序，2.拓扑排序可能有多种。

拓扑排序的解法模板：

DFS：用一个栈来保存所有搜索完成的节点，因为从一个节点开始，所有递归的点都是该点在拓扑排序的后任节点。 对于一个节点u，如果所有相邻节点都已经搜索完成，在回溯到u的时候，u也会变成一个已经搜索完成的节点，并且此时栈中的节点都是u的后任节点，将u放入栈中。 对图进行一边DFS，每个节点进行回溯的时候，把该节点放入栈中，最终从栈中得到的序列就是一个拓扑排序。

算法：图中节点状态为：未搜索、搜索中、已完成。

流程：

任取一个未搜索的节点进行DFS，假设为u，标记u为搜索中，并遍历该节点的每一个相邻节点v：

如果v是未搜索，那么开始搜索v，搜索完成后回溯到u

如果v为搜索中，那么存在环不存在拓扑排序

如果v为已完成，那么v已经在栈中，跳过该节点

当u的所有相邻节点都为已完成时，将u放入栈中，并标记为已完成。

BFS：很简单，面试当场写出来的可以。不同于DFS用栈保存逆向思维得到的序列，BFS顺序生成拓扑排序。排序最开始的节点一定是入度为0，声明一个队列，将所有入度为0的节点放入，并且将该节点的相邻节点入度减1，因为少了一个需要提前访问的节点。这样，如果相邻节点入度为0，就可以继续放入队列中，直到队列为空，即找到了一个满足条件的排列。或者图中包含环。

算法：使用队列，初始时所有入度为0的节点放入队列中，作为排列前面的节点并且顺序无所谓。

While 队列不为空：

取出队首节点u，将u放入排列中，所有u的相邻节点入度减1，如果某个相邻节点入度为0，放入队列中。

循环结束后，如果排列包含n个节点，则找到拓扑排序，否则存在环。