

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1Solution 2

```
1  # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3  # O(j + d) time | O(j + d) space
4  def topologicalSort(jobs, deps):
5      jobGraph = createJobGraph(jobs, deps)
6      return getOrderedJobs(jobGraph)
7
8
9  def createJobGraph(jobs, deps):
10     graph = JobGraph(jobs)
11     for prereq, job in deps:
12         graph.addPrereq(job, prereq)
13     return graph
14
15
16  def getOrderedJobs(graph):
17     orderedJobs = []
18     nodes = graph.nodes
19     while len(nodes):
20         node = nodes.pop()
21         containsCycle = depthFirstTraverse(node, orderedJobs)
22         if containsCycle:
23             return []
24     return orderedJobs
25
26
27  def depthFirstTraverse(node, orderedJobs):
28     if node.visited:
29         return False
30     if node.visiting:
31         return True
32     node.visiting = True
33     for prereqNode in node.prereqs:
34         containsCycle = depthFirstTraverse(prereqNode, orderedJobs)
35         if containsCycle:
36             return True
37     node.visited = True
38     node.visiting = False
39     orderedJobs.append(node.job)
40     return False
41
42
43  class JobGraph:
44     def __init__(self, jobs):
45         self.nodes = []
46         self.graph = {}
47         for job in jobs:
48             self.addNode(job)
49
50     def addPrereq(self, job, prereq):
51         jobNode = self.getNode(job)
52         prereqNode = self.getNode(prereq)
53         jobNode.prereqs.append(prereqNode)
54
55     def addNode(self, job):
56         self.graph[job] = JobNode(job)
57         self.nodes.append(self.graph[job])
58
59     def getNode(self, job):
60         if job not in self.graph:
61             self.addNode(job)
62         return self.graph[job]
63
64
65  class JobNode:
66     def __init__(self, job):
67         self.job = job
68         self.prereqs = []
69         self.visited = False
70         self.visiting = False
71
```