Solution 2

Solution 1

AlgoExpert **Quad Layout** Sublime Monokai 00:00:00 Go **12px**

Prompt Scratchpad Our Solution(s) **Video Explanation**

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
   package main
 5 type Dep struct {
     Prereq int
 8 }
10 // O(j + d) time | O(j + d) space
11 func TopologicalSort(jobs []int, deps []Dep) []int {
     jobGraph := createJobGraph(jobs, deps)
13
     return getOrderedJobs(jobGraph)
14 }
16 func createJobGraph(jobs []int, deps []Dep) *JobGraph {
17
      graph := NewJobGraph(jobs)
18
      for _, dep := range deps {
19
       graph.AddPrereq(dep.Job, dep.Prereq)
20
21
     return graph
22
23
24 func getOrderedJobs(graph *JobGraph) []int {
25
      \verb|orderedJobs| := [] int{|} |
      nodes := graph.Nodes
26
27
      for len(nodes) != 0 {
28
       node := nodes[len(nodes)-1]
29
       nodes = nodes[:len(nodes)-1]
       containsCycle := depthFirstTraverse(node, &orderedJobs)
30
31
       if containsCycle {
32
          return []int{}
33
34
35
      return orderedJobs
36 }
37
38
   func depthFirstTraverse(node *JobNode, orderedJobs *[]int) bool {
39
      if node.Visited {
40
       return false
41
      } else if node.Visiting {
       return true
43
44
      node.Visiting = true
45
      for _, prereqNode := range node.Prereqs {
46
       containsCycle := depthFirstTraverse(prereqNode, orderedJobs)
47
       if containsCycle {
48
         return true
49
50
51
      node.Visited = true
52
      node.Visiting = false
53
      *orderedJobs = append(*orderedJobs, node.Job)
54
      return false
55 }
56
57
   type JobGraph struct {
     Nodes []*JobNode
59
     Graph map[int]*JobNode
60
61
   func NewJobGraph(jobs []int) *JobGraph {
62
63
      g := &JobGraph{
64
       Graph: map[int]*JobNode{},
65
      for _, job := range jobs {
67
       g.AddNode(job)
68
69
      return g
70
71
72 func (g *JobGraph) AddPrereq(job, prereq int) {
73
      jobNode := g.GetNode(job)
74
      prereqNode := g.GetNode(prereq)
     jobNode.Prereqs = append(jobNode.Prereqs, prereqNode)
75
76 }
77
78 func (g *JobGraph) AddNode(job int) {
     g.Graph[job] = &JobNode{Job: job}
79
     g.Nodes = append(g.Nodes, g.Graph[job])
80
81 }
83 func (g *JobGraph) GetNode(job int) *JobNode {
84 if _, found := g.Graph[job]; !found {
     g.AddNode(job)
```

Run Code