AlgoExpert **Quad Layout** Go

Sublime 00:00:00 **12px** Monokai

Solution 1 Solution 2

Prompt

```
Scratchpad
              Our Solution(s)
                                 Video Explanation
```

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
    package main
   type Dep struct {
     Prereq int
8
   // 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
    func createJobGraph(jobs []int, deps []Dep) *JobGraph {
16
17
      graph := NewJobGraph(jobs)
18
      for _, dep := range deps {
19
       graph.AddDep(dep.Prereq, dep.Job)
20
21
     return graph
22
24
    func getOrderedJobs(graph *JobGraph) []int {
25
      orderedJobs := []int{}
      nodesWithNoPrereqs := []*JobNode{}
26
27
      for \_, node := range graph.Nodes {
28
        if node.NumOfPrereqs == 0 {
29
          nodesWithNoPrereqs = append(nodesWithNoPrereqs, node)
30
31
32
      for len(nodesWithNoPrereqs) > 0 {
        node := nodesWithNoPrereqs[len(nodesWithNoPrereqs)-1]
33
34
        nodesWithNoPrereqs = nodesWithNoPrereqs[:len(nodesWithNoPrereqs)-1]
35
        orderedJobs = append(orderedJobs, node.Job)
36
        removeDeps(node, &nodesWithNoPrereqs)
37
      for _, node := range graph.Nodes {
38
39
        if node.NumOfPrereqs > 0 {
40
          return []int{}
41
43
      return orderedJobs
44
45
    func removeDeps(node *JobNode, nodesWithNoPrereqs *[]*JobNode) {
46
47
      for len(node.Deps) > 0 {
48
        dep := node.Deps[len(node.Deps)-1]
49
        node.Deps = node.Deps[:len(node.Deps)-1]
50
        dep.NumOfPrereqs--
        if dep.NumOfPrereqs == 0 {
51
52
          *nodesWithNoPrereqs = append(*nodesWithNoPrereqs, dep)
53
54
55
56
57
    type JobGraph struct {
      Nodes []*JobNode
59
      Graph map[int]*JobNode
60
61
62
    func NewJobGraph(jobs []int) *JobGraph {
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) AddDep(job, dep int) {
73
      jobNode, depNode := g.GetNode(job), g.GetNode(dep)
74
      jobNode.Deps = append(jobNode.Deps, depNode)
      depNode.NumOfPrereqs++
75
76
78
    func (g *JobGraph) AddNode(job int) {
79
      g.Graph[job] = &JobNode{Job: job}
      g.Nodes = append(g.Nodes, g.Graph[job])
81
83 func (g *JobGraph) GetNode(job int) *JobNode {
84 if _, found := g.Graph[job]; !found {
      g.AddNode(job)
```

Run Code