

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type Dep struct {
6     Prereq int
7     Job    int
8 }
9
10 // O(j + d) time | O(j + d) space
11 func TopologicalSort(jobs []int, deps []Dep) []int {
12     jobGraph := createJobGraph(jobs, deps)
13     return getOrderedJobs(jobGraph)
14 }
15
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     orderedJobs := []int{}
26     nodes := graph.Nodes
27     for len(nodes) != 0 {
28         node := nodes[len(nodes)-1]
29         nodes = nodes[:len(nodes)-1]
30         containsCycle := depthFirstTraverse(node, &orderedJobs)
31         if containsCycle {
32             return []int{}
33         }
34     }
35 }
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 type Dep struct {
4     Prereq int
5     Job    int
6 }
7
8 func TopologicalSort(jobs []int, deps []Dep) []int {
9     // Write your code here.
10    return nil
11 }
12
```

Our Tests

```
1 // Test 1
2 // Pass
3 // Test 2
4 // Pass
5 // Test 3
6 // Pass
7 // Test 4
8 // Pass
9 // Test 5
10 // Pass
11 // Test 6
12 // Pass
13 // Test 7
14 // Pass
15 // Test 8
16 // Pass
17 // Test 9
18 // Pass
19 // Test 10
20 // Pass
21 // Test 11
22 // Pass
23 // Test 12
24 // Pass
25 // Test 13
26 // Pass
27 // Test 14
28 // Pass
29 // Test 15
30 // Pass
31 // Test 16
32 // Pass
33 // Test 17
34 // Pass
35 // Test 18
36 // Pass
37 // Test 19
38 // Pass
39 // Test 20
40 // Pass
41 // Test 21
42 // Pass
43 // Test 22
44 // Pass
45 // Test 23
46 // Pass
47 // Test 24
48 // Pass
49 // Test 25
50 // Pass
51 // Test 26
52 // Pass
53 // Test 27
54 // Pass
55 // Test 28
56 // Pass
57 // Test 29
58 // Pass
59 // Test 30
60 // Pass
61 // Test 31
62 // Pass
63 // Test 32
64 // Pass
65 // Test 33
66 // Pass
67 // Test 34
68 // Pass
69 // Test 35
70 // Pass
71 // Test 36
72 // Pass
73 // Test 37
74 // Pass
75 // Test 38
76 // Pass
77 // Test 39
78 // Pass
79 // Test 40
80 // Pass
81 // Test 41
82 // Pass
83 // Test 42
84 // Pass
85 // Test 43
86 // Pass
87 // Test 44
88 // Pass
89 // Test 45
90 // Pass
91 // Test 46
92 // Pass
93 // Test 47
94 // Pass
95 // Test 48
96 // Pass
97 // Test 49
98 // Pass
99 // Test 50
100 // Pass
```

Custom Output

Submit Code

```
1 // Custom Output
2 // Submit Code
```

