

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.AddDep(dep.Prereq, dep.Job)
20     }
21     return graph
22 }
23
24 func getOrderedJobs(graph *JobGraph) []int {
25     orderedJobs := []int{}
26     nodesWithNoPrereqs := []*JobNode{}
27     for _, node := range graph.Nodes {
28         if node.NumOfPrereqs == 0 {
29             nodesWithNoPrereqs = append(nodesWithNoPrereqs, node)
30         }
31     }
32     for len(nodesWithNoPrereqs) > 0 {
33         node := nodesWithNoPrereqs[len(nodesWithNoPrereqs)-1]
```

Our Tests

```
1 // Test 1
2
3 func TestTopologicalSort() {
4     jobs := []int{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
5     deps := []Dep{
6         {1, 2}, {1, 3}, {2, 4}, {3, 4}, {4, 5}, {5, 6}, {6, 7}, {7, 8}, {8, 9}, {9, 10}
7     }
8     expected := []int{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
9     actual := TopologicalSort(jobs, deps)
10    assert.Equal(t, expected, actual)
```

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
```

Custom Output

Submit Code

```
1
2
3
4
5
6
7
8
9
10
```

