Our Solution(s) Run Code

Solution 1 Solution 2

```go
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

package main

type Dep struct {
  Prereq int
  Job    int
}

// O(j + d) time | O(j + d) space
func TopologicalSort(jobs []int, deps []Dep) []int {
  jobGraph := createJobGraph(jobs, deps)
  return getOrderedJobs(jobGraph)
}

func createJobGraph(jobs []int, deps []Dep) *JobGraph {
  graph := NewJobGraph(jobs)
  for _, dep := range deps {
    graph.AddPrereq(dep.Job, dep.Prereq)
  }
  return graph
}

func getOrderedJobs(graph *JobGraph) []int {
  orderedJobs := []int{}
  nodes := graph.Nodes
  for len(nodes) != 0 {
    node := nodes[len(nodes)-1]
    nodes = nodes[:len(nodes)-1]
    containsCycle := depthFirstTraverse(node, &orderedJobs)
    if containsCycle {
      return []int{}
    }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

type Dep struct {
  Prereq int
  Job    int
}

func TopologicalSort(jobs []int, deps []Dep) []int {
  // Write your code here.
  return nil
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.