

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     class JobNode {
5         let job: Int
6         var prerequisites: [JobNode]
7         var visited: Bool
8         var visiting: Bool
9
10        init(job: Int) {
11            self.job = job
12            prerequisites = [JobNode]()
13            visited = false
14            visiting = false
15        }
16    }
17
18    class JobGraph {
19        var nodes: [JobNode]
20        var graph: [Int: JobNode]
21
22        init(jobs: [Int]) {
23            nodes = [JobNode]()
24            graph = [Int: JobNode]()
25            for job in jobs {
26                addNode(job: job)
27            }
28        }
29
30        func addNode(job: Int) {
31            let jobNode = JobNode(job: job)
32
33            nodes.append(jobNode)
```

Solution 1Solution 2Solution 3

```
1 class Program {
2     func topologicalSort(jobs: [Int], dependencies: [[Int]]) -> [Int] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     func topologicalSort(jobs: [Int], dependencies: [[Int]]) -> [Int] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

```
1 class Program {
2     func topologicalSort(jobs: [Int], dependencies: [[Int]]) -> [Int] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

