Our Solution(s) Run Code

Solution 1 Solution 2

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

// O(j + d) time | O(j + d) space
function topologicalSort(jobs, deps) {
  const jobGraph = createJobGraph(jobs, deps);
  return getOrderedJobs(jobGraph);
}

function createJobGraph(jobs, deps) {
  const graph = new JobGraph(jobs);
  for (const [prereq, job] of deps) {
    graph.addPrereq(job, prereq);
  }
  return graph;
}

function getOrderedJobs(graph) {
  const orderedJobs = [];
  const {nodes} = graph;
  while (nodes.length) {
    const node = nodes.pop();
    const containsCycle = depthFirstTraverse(node, orderedJobs);
    if (containsCycle) return [];
  }
  return orderedJobs;
}

function depthFirstTraverse(node, orderedJobs) {
  if (node.visited) return false;
  if (node.visiting) return true;
  node.visiting = true;
  for (const prereqNode of node.prereqs) {
    const containsCycle = depthFirstTraverse(prereqNode, orderedJobs);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function topologicalSort(jobs, deps) {
  // Write your code here.
}

// Do not edit the line below.
exports.topologicalSort = topologicalSort;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.