Our Solution(s) Run Code

Solution 1 Solution 2

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

import java.util.*;

class Program {
  // O(j + d) time | O(j + d) space
  public static List<Integer> topologicalSort(List<Integer> jobs, List
    JobGraph jobGraph = createJobGraph(jobs, deps);
    return getOrderedJobs(jobGraph);
  }

  public static JobGraph createJobGraph(List<Integer> jobs, List<Integ
    JobGraph graph = new JobGraph(jobs);
    for (Integer[] dep : deps) {
      graph.addDep(dep[0], dep[1]);
    }
    return graph;
  }

  public static List<Integer> getOrderedJobs(JobGraph graph) {
    List<Integer> orderedJobs = new ArrayList<Integer>();
    List<JobNode> nodesWithNoPrereqs = new ArrayList<JobNode>();
    for (JobNode node : graph.nodes) {
      if (node.numOfPrereqs == 0) {
        nodesWithNoPrereqs.add(node);
      }
    }
    while (nodesWithNoPrereqs.size() > 0) {
      JobNode node = nodesWithNoPrereqs.get(nodesWithNoPrereqs.size()
      nodesWithNoPrereqs.remove(nodesWithNoPrereqs.size() - 1);
      orderedJobs.add(node.job);
      removeDeps(node, nodesWithNoPrereqs);
    }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
import java.util.*;

class Program {
  public static List<Integer> topologicalSort(List<Integer> jobs, List
    // Write your code here.
    return null;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.