CourseSchedule.java

```
1     package Graph;
2
3     import java.util.ArrayList;
4     import java.util.List;
5
6     public class CourseSchedule {
7         public static int[] findOrder(int n, int[][] prerequisites) {
8             List<Integer>[] adjList = new List[n];;
9  2          for(int i=0; i<n; i++) {
10                adjList[i] = new ArrayList<>();
11            }
12            for(int[] prerequisite : prerequisites) {
13                adjList[prerequisite[0]].add(prerequisite[1]);
14            }
15            List<Integer> res = new ArrayList<>();
16            boolean[] visiting = new boolean[n];
17            boolean[] visited = new boolean[n];
18 2          for(int i=0; i<n; i++) {
19 1              if(!dfs(adjList, i, visiting, visited, res)) {
20 1                  return new int[0];
21                }
22            }
23            int[] resArray = new int[res.size()];
24 2          for(int i=0; i<res.size(); i++){
25                resArray[i] = res.get(i);
26            }
27 1          return resArray;
28        }
29
30        public static boolean dfs(List<Integer>[] adjList, int curr, boolean[] visiting, boolean[] visited, List<Integer>
31            // node has already been visited and added to res
32 1          if(visited[curr]) {
33 1              return true;
34            }
35            // we are encoutering a node already on the path, aka there is a cycle
36 1          if(visiting[curr]) {
37 1              return false;
38            }
39            visiting[curr] = true;
40            for(int neighbour : adjList[curr]) {
41 1              if(!dfs(adjList, neighbour, visiting, visited, res)) {
42 1                  return false;
43                }
44            }
45            visiting[curr] = false;
46            visited[curr] = true;
47            res.add(curr);
48 1          return true;
49        }
50    }
```

Mutations

```
9   1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
18  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
19  1. negated conditional → KILLED
20  1. replaced return value with null for Graph/CourseSchedule::findOrder → KILLED
24  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
27  1. replaced return value with null for Graph/CourseSchedule::findOrder → KILLED
32  1. negated conditional → KILLED
33  1. replaced boolean return with false for Graph/CourseSchedule::dfs → KILLED
36  1. negated conditional → KILLED
37  1. replaced boolean return with true for Graph/CourseSchedule::dfs → KILLED
41  1. negated conditional → KILLED
42  1. replaced boolean return with true for Graph/CourseSchedule::dfs → SURVIVED
48  1. replaced boolean return with false for Graph/CourseSchedule::dfs → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- Graph.CourseScheduleTest.testFindOrder(Graph.CourseScheduleTest) (1 ms)

Report generated by PIT 1.15.0