Is there a Route between Two Nodes in a Graph?

Shikha Nair

October 2, 2017

Question

Given a directed graph and two nodes, design an algorithm to find out whether there is a route between the two nodes.

Explanation and Algorithm

The solution to this problem is rather simple: use any graph traversal method such as breadth-first or depth-first. Starting with one node, do a search and check if we find the second node. In order to avoid repetition and cycles, we should mark each node we find as "visited".

Hints

1. Since you are given a starting and ending point, is there a way to see if a directed graph has a path between the two?
2. Think about graph traversal. Would this be an effective way to find a potential path?
3. What is one way you can keep your checks form being repetitive or cyclic? In other words, can we find a way to see if we have already seen the current node?

Code

```
public enum State {
   Unvisited, Visited, Visiting;
}

 public static boolean search(Graph g, Node start, Node end) {
    LinkedList<Node> q = new LinkedList<Node>(); // operates as Stack
```

```
    for (Node u : g.getNodes()) {
      u.state = State.Unvisited;
    }
  start.state = State.Visiting;
  q.add(start);
  Node u;
  while(!q.isEmpty()) {
     u = q.removeFirst(); // i.e., pop()
     if (u != null) {
        for (Node v : u.getAdjacent()) {
           if (v.state == State.Unvisited) {
              if (v == end) {
                 return true;
              } else {
                v.state = State.Visiting;
                q.add(v);
              }
           }
        }
        u.state = State.Visited;
     }
  }
  return false;
}
```

Big O analysis

Since, in the worst case, this method 'reaches" each node in the graph, the run-time is O(n).

Sources

Question, answer and other material taken from Cracking the Coding Interview 4th edition by Gayle Laakmann McDowell.