

Study problems for midterm 1

CSE 101

1. Exercise 3.11. The algorithm can be written in just two lines, by calling **explore** as a subroutine.
2. Exercise 3.15. This is about taking a real-world problem and relating it to an abstract graph problem.
3. Exercise 3.22. You should be able to do this in linear time.
4. Exercise 3.24. It helps if you first compute a certain ordering of the nodes.
5. Exercise 3.25. This is a typical situation where you want to solve a problem for general directed graphs, but you first solve it for DAGs and then extend your solution by thinking about strongly connected components.
6. Exercise 3.26(a). You should try writing this out carefully.
7. Exercise 4.4. Being able to construct counterexamples is an important part of algorithm design, because most of the algorithms one first thinks up are incorrect.
8. Exercise 4.5. I recommend first running BFS and then doing a second pass through the graph.
9. Exercise 4.11. Does Dijkstra's algorithm help with this?
10. Exercise 4.15. You could do this by modifying Dijkstra's algorithm, but if you do so, you will have to prove that your modified version is correct. Therefore, in terms of proving correctness, it is simpler to simply call Dijkstra's algorithm as a subroutine.