

Homework 2

Released 2/8/2019

Due 2/18/2019 11:59pm in Gradescope

Instructions. You make work in groups, but you must write solutions yourself. List collaborators on your submission.

If you are asked to design an algorithm, please provide: (a) the pseudocode or precise description in words of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submissions. Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope.

1. **(25 points) Graph short answer.**

- (a) **(10 points) K&T Ch3.Ex7.** Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected. Decide whether you think the claim is true or false, and either give a proof of the claim or give a counterexample.
- (b) **(15 points) K&T Ch3.Ex9.** Let $G = (V, E)$ be an n node undirected graph containing two nodes s and t , such that the distance between s and t is strictly greater than $n/2$. Show that there must be some node v , not equal to either s or t such that deleting v from G destroys all $s - t$ paths. In other words, the graph G' obtained by deleting v contains no paths from s to t . Hint: consider a BFS starting at some node. Think about the layers.

2. **(25 points) Directed Graphs.** Given a directed acyclic graph G , give a linear time algorithm to determine if the graph has a directed path that visits every vertex.

3. **(25 points) Visiting all edges.**

Suppose you have a connected network of two-way streets. Show that you can drive along these streets so that you visit all streets and you drive along each side of every street exactly once. Further, show that you can do this such that, at each intersection, you do not leave by the street you **first** used to enter that intersection unless you have previously left via all other streets from that intersection.

4. **(25 points) K&T Ch3 Ex10** A number of art museums around the country have been featuring work by an artist named Mark Lombardi (1951-2000), consisting of a set of intricately rendered graphs. Building on a great deal of research, these graphs encode the relationships among people involved in major political scandals over the past several decades: the nodes correspond to participants, and each edge indicates some type of relationship between a pair of participants. And so, if you peer closely enough at the drawings, you can trace out ominous-looking paths from a high-ranking U.S. government official, to a former business partner, to a bank in Switzerland, to a shadowy arms dealer.

Such pictures form striking examples of social networks, which, as we discussed in Section 3.1, have nodes representing people and organizations, and edges representing relationships of various kinds. And the short paths that abound in these networks have attracted considerable attention recently, as people ponder what they mean. In the case of Mark Lombardi's graphs, they hint at the short set of steps that can carry you from the reputable to the disreputable.

Of course, a single, spurious short path between nodes v and w in such a network may be more coincidental than anything else; a large number of short paths between v and w can be much more convincing. So in addition to the problem of computing a single shortest $v - w$ path in a graph G , social networks researchers have looked at the problem of determining the number of shortest $v - w$ paths. This turns out to be a problem that can be solved efficiently. Suppose we are given an undirected graph $G = (V, E)$, and we identify two nodes v and w in G . Give an algorithm that computes the number of shortest $v - w$ paths in G . (The algorithm should not list all the paths; just the number suffices.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges.

5. **(0 points).** How long did it take you to complete this assignment?