CS420/520: Graph Theory with Applications to CS, Winter 2020

Homework 1

Due: Thursday, 1/23/20

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.

Readings:

(A) Jeff lecture notes on basic graph algorithms: http://jeffe.cs.illinois.edu/teaching/algorithms/book/05-graphs.pdf.

(B) Jeff lecture notes on basic graph algorithms: http://jeffe.cs.illinois.edu/teaching/algorithms/book/06-dfs.pdf.

Problems for practice.

- Problems (1), (3), (7), (12) from (A).

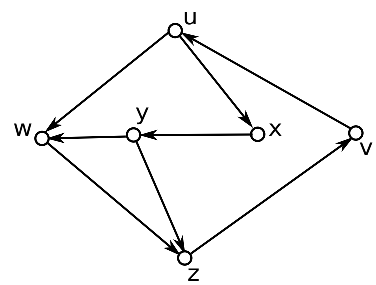
Problem 1. A graph $G = (V, E)$ is bipartite if the vertices V can be partitioned into two subsets L and R, such that every edge has one vertex in L and the other in R.

(a) Prove that every tree is a bipartite graph.

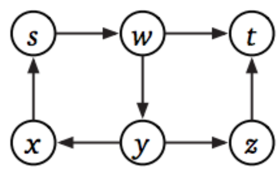
(b) Prove that a graph G is bipartite if and only if every cycle in G has an even number of edges.

(c) Describe and analyze an efficient algorithm that determines whether a given undirected graph is bipartite.

Problem 2. Draw all possible DFS trees rooted at x for the following graph. Mark, tree, forward, backward, and cross edges.



Problem 3. Suppose you are given a directed graph $G = (V, E)$ and two vertices s and t. Describe and analyze an algorithm to determine if there is a walk in G from s to t (possibly repeating vertices and/or edges) whose length is divisible by 3. For example, given the graph shown below, with the indicated vertices s and t, your algorithm should return $True$, because the walk $s \to w \to y \to x \to s \to w \to t$ has length 6.



Problem 4. Let $G = (V, E)$ be an undirected unweighted graph, and let $s, t \in V$. Show that at least one of the following conditions hold.

1. The distance between s and t is at most $\sqrt{V} + 1$.
2. There is a subset $S \subset V$ of cardinality at most $\sqrt{V}$ whose removal disconnects s and t. (Hint: Use the fact that the BFS tree does not have skip edges.)

Problem 5. A directed graph G is semi-connected if, for every pair of vertices u and v, either u is reachable from v or v is reachable from u (or both).

(a) Give an example of a directed acyclic graph with a unique source that is not semi-connected.

(b) Describe and analyze an algorithm to determine whether a given directed acyclic graph is semi-connected.