CSE421: Design and Analysis of Algorithms	October 17, 2020

Homework 3

Anup Rao

Due: October 23, 2020

Read the fine print[1]. An algorithm is said to run in polynomial time if it runs in time $O(n^d)$ for some constant d on inputs of size n. Each problem is worth 10 points:

1. Give a polynomial time algorithm that takes an undirected graph with m edges as input and outputs a coloring of the vertices with 3 colors, so that at least $2m/3$ of the edges are properly colored. An edge is properly colored if its vertices get distinct colors. HINT: Give a greedy algorithm that colors each vertex one by one.

2. You are given a directed graph G on n vertices, with m edges. The edges have (possibly negative) weighted edges, and the graph is promised to have no negative cycles. In addition, you are given a rooted tree T with root s. T is promised to have exactly one path from s to every other vertex in the graph. Give an $O(m + n)$ time algorithm to decide whether or not T is a valid shortest path tree for G.

3. In class we discussed an algorithm to color the vertices of an undirected n vertex graph with 2 colors so that every edge gets exactly 2 colors (assuming such a coloring exists). We know of no such algorithm for finding 3-colorings in polynomial time. Here we'll figure out how to color a 3-colorable graph with $O(\sqrt{n})$ colors.

   (a) Give a greedy polynomial time algorithm that can properly color the vertices with $\Delta + 1$ colors, as long as every vertex of the graph has degree at most Δ.

   (b) Give a polynomial time algorithm that can properly color the graph with $O(\sqrt{n})$ colors, as long as the input graph is promised to be 3-colorable. HINT: If a vertex v has more than $\sqrt{n}$ neighbors, then argue that the subgraph of the neighbors of v must be bipartite, and use the algorithm from class to color v and its neighbors with 3 new colors. Continue this process until every vertex has less than $\sqrt{n}$ neighbors, and then use the algorithm from part (a).

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.