Homework 3 Due: 23:59:59, May 3, 2023

Problem 1 (2 pts) Given a weighted graph $G = (V, E)$, let T be its shortest-path tree from a node s in V. Suppose we add a positive value c to all weights of edges in E. Then is T still the shortest-path tree of the resulting graph? Prove or disprove this claim.

Problem 2 Let T be a rooted binary tree with n nodes with root node r. Let G be the directed graph constructed from T by replacing each edge $(p(u), u)$ of T with a directed edge from $p(u)$ to u, and by adding for each leaf node w a directed edge from w to r, where $p(u)$ is the parent node of u. All edge weights in G are positive.

(a) (2 pts) How much time does it take to compute the shortest path from node u to node v in G if you use Dijkstra's algorithm?

(b) (3 pts) Can you do it faster? If so, describe your algorithm and analyze its correctness and running time.

Problem 3 (4 pts) Suppose you are given a directed graph G in which every edge has a negative weight, and a source vertex s. Describe and analyze an efficient algorithm that computes the shortest-path distances from s to every other vertex in G. Specifically, for every vertex t:

- If t is not reachable from s, your algorithm should report $dist(t) = \infty$.
- If G has a cycle that is reachable from s, and t is reachable from that cycle, then the shortest-path distance from s to t is not well-defined, because there are paths from s to t of arbitrarily large negative length. In this case, your algorithm should report $dist(t) = -\infty$.
- If neither of the two previous conditions applies, your algorithm should report the correct shortest-path distance from s to t.

Problem 4 (3 pts) Given an undirected connected graph $G = (V, E)$, a coloring of G using k colors is an assignment of the k colors to vertices, one color to each vertex, such that no two adjacent vertices are colored using the same color. Design a greedy algorithm that achieves a coloring of G using $d + 1$ colors, where d is the maximum degree of a vertex in G. Analyze the time complexity and correctness of your algorithm.

Problem 5 (4 pts) Given a set S of n intervals $I_1, I_2, \ldots, I_n$ on the real line, we want to find a smallest subset T of the intervals such that any real value contained in an interval in S is also contained in some interval in T. Describe an efficient algorithm for this problem and analyze the correctness and running time.

Problem 6 (4 pts) Let X be a set of n intervals on the real line. We say that a set P of points *stabs* X if every interval in X contains at least one point in P. Describe and analyze an efficient algorithm to compute the smallest set of points that stabs X. Assume that your input consists of two arrays $L[1..n]$ and $R[1..n]$, representing the left and right endpoints of the intervals in X. If you use a greedy algorithm, you must prove that it is correct. (In the following figure, intervals are stabbed by three points.)