Fundamental Algorithms

Homework 6

Due: July 19th, 5:30 PM EST

Instructions

Please answer each **Problem** on a separate page. Submissions must be uploaded to your account on Gradescope by the due date and time above.

Problems To Submit

Problem 1 (35 points)

Recall that a cycle in an undirected graph is a sequence of distinct vertices $(v_1, v_2, \ldots, v_k)$ with $k \geq 3$ such that the edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}$, and $\{v_k, v_1\}$, all belong to the graph.
Given an undirected connected graph $G = (V, E)$, develop an algorithm that determines whether G has a cycle. Your algorithm must run in $O(|V| + |E|)$ time.

Problem 2 (15+15 points)

Provide counterexamples to each of the following statements.

(a) If a directed graph G contains a path from u to v, and if $u.d < v.d$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced.
Note that a forest is a union of some disjoint trees.

(b) If a directed graph G contains a path from u to v, then any depth-first search on G must result in $v.d \leq u.f$.

Problem 3 (35 points)

Given a directed graph and two vertices u, v in the graph, the function P finds the shortest path from u to v in that graph. We have access to the function P and want to use it to solve the following problem.

Consider the directed graph G such that each edge of G is colored by red or blue. Given two vertices s, t in G, we want to find the shortest path from s to t in G which is *color-separable*. In other words, the path must first consist of some red edges (possibly none of them) followed by some blue edges (possibly none of them). Thus, once we encounter a blue edge in the path, the following edges in the path must be also blue.

Develop an $O(|V|+|E|+T_P)$-time algorithm to find the shortest color-separable path from s to t in G, where T_P denotes the running time of function P. Your algorithm must use the function P once on a carefully constructed graph. Note that you should not use BFS or explore the graph yourself. Instead, you must call the function P.
Justify why the running time of your algorithm satisfies the required bound.

Hint: First, consider the graph with only the red edges of G. Then, restrict yourself to the graph with only the blue edges of G. Can you combine these two graphs somehow to construct a new graph and solve the problem?

Practice Problems

Problem 1

Let G be a directed graph. After running DFS algorithm on G, the vertex v of G ended up in a DFS tree containing only v, even though v has both incoming and outgoing edges in G. Fully explain how this happened.

Problem 2

Let $G = (V, E)$ be an undirected connected graph where each edge has a weight from the set $\{1, 10, 25\}$. Describe an $O(|V| + |E|)$ time algorithm to find an MST of G.