CSE 101 Homework 2

Spring 2018

This homework is due on gradescope Friday April 27th at 11:59pm. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in LaTeX is recommended though not required.

Question 1 (BFS and DFS trees, 25 points). Let $G$ be a connected, undirected graph containing some vertex $s$. Suppose that BFS and DFS are both run on $G$ starting at $s$ and that the breadth first search and depth first search trees produced are the same. Show that $G$ is a tree (or just that $G$ has no edges other than those in the BFS/DFS tree).

Question 2 (Shortest Paths with Interest, 50 points). Bob recently inherited a large sum of money. He lives in sourceville and needs to travel to sinktown in order to claim the money. Unfortunately, travel is very expensive in his country. To make matters worse, he doesn’t have the money on hand to pay for the trip. He could take a loan, but it will accrue interest rapidly the longer his travels take him. Bob’s country is represented by an undirected graph $G$, with sourceville being a vertex $s$ and sinktown being represented by another vertex $t$. Each edge $(u, v)$ of $G$ takes exactly one day to traverse, but would cost Bob a total of $c(u, v) \geq 0$ to traverse. Furthermore, due to interest payments, the total cost of travelling along a path $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k$ is

$$C(v_0, v_1, \ldots, v_k) = \sum_{h=1}^{k} \left( 1.1^h c(v_{k-h}, v_{k-h+1}) \right).$$

(a) Give an $O((|V| + |E|) \log(|V|))$ time algorithm that given $G, s, t$ and $c$ computes the cost of the least expensive (including interest) path from $s$ to $t$. [Hint: You may want to define a distance function where $d(w) = \min_{(u, w) \in E} (1.1)(d(u) + c(u, w))$.] [30 points]

(b) Your algorithm will likely no longer work if the interest rate were negative and, for example,

$$C(v_0, v_1, \ldots, v_k) = \sum_{h=1}^{k} \left( 0.9^h c(v_{k-h}, v_{k-h+1}) \right).$$

What goes wrong with your above analysis here? [20 points]

Question 3 (Collecting Special Vertices, 25 points). Given a DAG $G$ and a set $S \subset V$ of its vertices, give a linear time algorithm to find the path in $G$ that contains the greatest possible number of vertices of $S$. [Hint: First topologically sort $G$ and then solve the problem for each starting vertex in some order.]

Question 4 (Extra credit, 1 point). Approximately how much time did you spend working on this homework?