

CS 256: Algorithm Design and Analysis

Assignment 2 (due 02/21/2024)

Instructor: Sam McCauley

All problems on this assignment are worth 10 points. (Bonus and extra credit problems are in addition to these totals, and their point values are marked separately.)

For this assignment, you are allowed to work in groups of 2 or 3. If you do, please write the names of the students in this group here:

- Student 1
- Student 2
- Student 3

Depth-First Search & Traversal Applications

Problem 1. (KT 3.5) Let $G = (V, E)$ be a connected undirected graph, and let T be a depth-first spanning tree of G rooted at some node v . Prove that if T is also a breadth-first spanning tree of G rooted at v , then $E = T$, that is, all edges of G must be present in T and vice versa.

Hint. Use the properties of BFS and DFS that we proved in class.

Solution.

□

Problem 2. Recall that the diameter of a graph G is the “longest shortest path”, that is, $\text{diam}(G) = \max\{\text{dist}(u, v) : u, v \in V\}$, where $d(u, v)$ is the length of the shortest path from u to v in G . Let $T = (V, E)$ be a tree with n vertices. In this question, we will design an $O(n)$ -time algorithm to find the diameter of T by modifying recursive DFS.¹

In particular, suppose we modify recursive DFS so that it also computes, for each vertex v : (i) the diameter of the subtree of T rooted at v , and (ii) the longest path from v to a leaf in the subtree of T rooted at v .

- (a) Why do you think we need to know both (i) and (ii)?

Solution.

□

- (b) Show how, knowing this information for all children of some vertex u , we can determine this information for u itself.

Solution.

□

- (c) Argue why a modified recursive DFS (that computes the diameter as explained in part(b)) would still be $O(n)$ time. (To analyze the running time, you’d have describe how the algorithm works but only in so much detail as to justify the time bound.)

Solution.

□

¹It might seem odd to apply BFS or DFS to a tree—you just get the tree you started with! However, the traversal of a tree in a particular order can allow for efficient computation of useful quantities.

Problem 3. (KT 3.12) You're helping a group of ethnographers analyze some oral history data they've collected by interviewing members of a village to learn about the lives of people who have lived there over the past two hundred years. From these interviews, they've learned about a set of n people (all of them now deceased), whom we'll denote P_1, P_2, \dots, P_n . They've also collected facts about when these people lived relative to one another. Each fact has one of the following two forms:

- For some i and j , person P_i died before person P_j was born; or
- For some i and j , the life spans of P_i and P_j overlapped at least partially.

Naturally, they're not sure that all these facts are correct; memories are not so good, and a lot of this was passed down by word of mouth. So what they'd like you to determine is whether the data they've collected is at least internally consistent, in the sense that there could have existed a set of people for which all the facts they've learned simultaneously hold.

Give an algorithm to do this: either it should produce proposed dates of birth and death for each of the n people so that all the facts hold true, or it should report (correctly) that no such dates can exist—that is, the facts collected by the ethnographers are not internally consistent. You can assume that the “dates” have the form $1, 2, \dots$

Justify the correctness and running time of your algorithm.

Hint. Model the collected facts as a directed graph. The direction of an edge (x, y) can represent an event x occurring before event y . What should be the nodes of this graph?

Solution.

□

Problem 4 (Finding Bridges). Let $G = (V, E)$ be a connected undirected graph. A *bridge* is an edge $e \in E$ whose deletion disconnects the graph G . In this question, we will design an $O(n + m)$ -time algorithm to determine if G contains a bridge.

- (a) Consider a DFS tree T of $G = (V, E)$. Can an edge $e \in E \setminus T$ be a bridge?

Solution.

□

- (b) Give a necessary and sufficient condition under which an edge $e = (u, v) \in T$ is a bridge of G .

Solution.

□

- (c) (**Extra credit (5 points)**) Modify depth-first search to determine whether a given graph G contains a bridge in $O(n + m)$ time. *Solution.*

□

Bonus Feedback Question.

This question is optional and intended to guide future problem sets.

Problem 5. (2 points) This problem set took me:

- (a) Between 6-8 hours.
- (b) Between 4-6 hours.
- (c) Between 8-10 hours.
- (d) Other (please specify)

Solution.

□