

CS180 Homework 2

Due: 11:59 pm PST, May 3

Note that for all the questions below we assume m (number of edges) $\geq n$ (number of nodes) so you don't need to distinguish between $O(m)$ and $O(m + n)$.

1. (10 pt) A binary tree is a rooted tree in which each node has at most two children. Prove that in any binary tree the number of nodes with two children is exactly one less than the number of leaves. (Hint: prove by induction.)
2. (25 pt) For an undirected and unweighted graph, the BFS algorithm introduced in the class and textbook will output the shortest path from source (node s) to other nodes. Modify the BFS algorithm to also output number of shortest paths from s to other node. Write down the pseudo code and explain why it's correct. We assume the graph is connected and your algorithm should take $O(m)$ time.
3. (30 pt) We define the distance between two nodes u, v in a graph as the minimum number of edges in a path joining them, denoted by $\text{dist}(u, v)$. The diameter of graph $G = (V, E)$ is the maximum distance between any pair of nodes, i.e.,

$$\text{diameter}(G) = \max_{u, v \in V} \text{dist}(u, v)$$

- (a) Design an algorithm that runs in time $O(nm)$ and finds the diameter of G .
- (b) If the graph is a tree, denoted as $T = (V, E)$, the diameter can be computed efficiently. Please give an $O(n)$ time algorithm¹ to find the diameter of T . Write down the pseudo code and explain why your algorithm is correct.

Suggestion: consider modifying the DFS so that it can compute the following two numbers for each node v :

- d_v : the diameter of the subtree of T rooted at v ,
- ℓ_v : the longest path from v to a leaf in the subtree of T rooted at v .

Think about how to compute these numbers in DFS recursion and how to compute the diameter of T using these numbers.

4. In this problem, we consider another notion of “semi-connectivity”. A directed graph is semi-connected if, for all pairs of $u, v \in V$, there is a path from u to v or from v to u . We will develop an algorithm to test if a directed graph $G = (V, E)$ is semi-connected.
 - (a) (15 pt) Consider a simplified case where G is a DAG. Design an $O(m)$ time algorithm to test whether a DAG is semi-connected.
 - (b) (5 pt) For a directed graph G , we can construct a pseudo-graph G' by the following way:
 - Decompose G into a set of Strongly Connected Components (SCCs). This can be done using the Kosaraju's algorithm described in the class. The output of Kosaraju's algorithm will be k node sets S_1, \dots, S_k , where each S_i contains nodes in the i -th SCC. The sets will be maximal in the sense that for each S_i , adding any other node will break its strong connectivity (we didn't officially prove this in the class but you can directly use this property).
 - Take each SCC as a node to create the pseudo-graph G' . More specifically, $G' = (V', E')$ where V' has K nodes v'_1, \dots, v'_k , each of them corresponds to a SCC; and $(v'_i, v'_j) \in E'$ if and only if there is an edge pointing from a node in S_i to a node in S_j in the original graph.

Prove that for any directed graph G , its pseudo-graph G' constructed this way will be a DAG.

¹Note that a tree with n nodes has $n - 1$ edges, so $O(n) = O(m)$

- (c) (15 pt) Design an algorithm (based on (a) and (b)) to test semi-connectivity of a directed graph in $O(m)$ time. Note that you can assume functions are given to conduct topological sort and to decompose a graph into SCCs (Kosaraju's algorithm). Prove the correctness of your algorithm.
-

- ★ Homework assignments are due on the exact time indicated. Please submit your homework using the Gradescope system. Email attachments or other electronic delivery methods are not acceptable. To learn how to use Gradescope, you can:
 - 1. Watch the one-minute video with complete instructions from here:
<https://www.youtube.com/watch?v=-wemzmvGPfg>
 - 2. Follow the instructions to generate a PDF scan of the assignments:
http://gradescope-static-assets.s3-us-west-2.amazonaws.com/help/submitting_hw_guide.pdf
 - 3. **Make sure you start each problem on a new page.**
- ★ We recommend to use \LaTeX , \LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receive fewer points.
- ★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.