02-713 Homework #3: Graph Traversals
Due: Feb. 7 by 9:30am

You may discuss these problems with your current classmates, but you must write up your solutions independently, without using common notes or worksheets. You must indicate at the top of your homework who you worked with. Your write up should be clear and concise. You are trying to convince a skeptical reader that your answers are correct. Your homework should be submitted via Autolab (`https://autolab.cs.cmu.edu/02713-s14/`) as a typeset PDF. A LaTeX tutorial and template are available on the class website if you choose to use that system to typeset. For problems asking for an algorithm: describe the algorithm, an argument why it is correct, and an estimation of how fast it will run. Use O notation for running times.

1. Let $G = (V, E)$ be an undirected graph, and let s and t be two vertices of G. Give an efficient algorithm for computing the *number* of different shortest paths from s to t. The length of a path is the number of edges on it, and two paths are different if the sets of edges that they use are different. You shouldn't return the actual paths, just their number.

   Hint 1: Store some information at each node.

   Hint 2: If your solution has a "$count + 1$" someplace in it, it's probably wrong.

2. (a) You are given a network of biking trails for a park in the form of an undirected graph $G = (V, E)$. Give an $O(|V|)$-time algorithm to check whether the park contains a set of trails that form a loop (i.e. a cycle: a path that you could ride around forever without turning around). Note that the running time of your algorithm should not depend on $|E|$, the number of trails. You don't need to output the cycle; just return YES or NO depending on whether it exists.

   (b) Suppose, to avoid collisions between bikes, the trails each have an enforced direction: you are allowed to go only the posted direction on a trail. Either argue that, with minor changes, your algorithm for part (a) still works in $O(|V|)$ time to find a biking loop or explain why it does not.

3. Use DFS to argue that if every vertex of an undirected graph G has at least $d \geq 2$ neighbors, then G contains a cycle of length at least d.

4. Let $G = (V, E)$ be a connected, undirected graph, and suppose $T_s = (V, F)$ is a tree on G created by exploring G using depth-first search starting from vertex s. Vertex s is the root of tree T_s. Argue that s has more than one child in T_s if and only if removing s from G breaks G into several disconnected parts.

5. The *diameter* of a connected, undirected graph $G = (V, E)$ is the length (in number of edges) of the *longest* shortest path between two nodes. (a) Show that if the diameter of a graph is d then there is some set $S \subseteq V$ with $|S| \leq n/(d-1)$ such that removing the vertices in S from the graph would break it into several disconnected pieces. (b) Give an efficient algorithm to find S.