CSCE629 Analysis of Algorithms

Homework 4

Texas A&M U, Fall 2019 *09/20/19*
Lecturer: Fang Song *Due: 10am, 09/27/19*

Instructions.

- Typeset your submission by LaTeX, and submit in PDF format. Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct, and you will get significantly more partial credit if you clearly identify the gap(s) in your solution. You may opt for the "I'll take 15%" option (details in Syllabus).
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources** for each problem. Be ready to explain your solutions orally to a course staff if asked.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

This assignment contains 4 questions, 5 pages for the total of 65 points and 10 bonus points. A random subset of the problems will be graded.

1. (Component graph) Given a directed graph $G = (V, E)$, we define another graph $G^{\text{SCC}} = (V^{\text{SCC}}, E^{\text{SCC}})$ called the *component graph* as follows. Suppose that G has strongly connected components $C_1, C_2, \ldots, C_k$. The vertex set V^{SCC} is $\{v_1, \ldots, v_k\}$ where $v_i \in C_i$. There is an edge $(v_i, v_j) \in E^{\text{SCC}}$ if G contains a directed edge $x \to y$ for some $x \in C_i$ and some $y \in C_j$. Alternatively, imagine contracting all edges whose incident vertices are within the same strongly connected component of G, and the resulting graph will be G^{SCC}.

   (a) (10 points) Prove or disapprove that G^{SCC} is a DAG.

   (b) (10 points (bonus)) Give an $O(|V| + |E|)$-time algorithm to compute the component graph of a directed graph $G = (V, E)$.

2. (10 points) (Singly connected graph) A directed graph $G = (V, E)$ is *singly* connected if G contains at most one simple (i.e. no vertex repeated) path from u to v for all vertices $u, v \in V$. Give an efficient algorithm to determine whether or not a directed graph is singly connected.

3. (Semi-connected graphs) A directed graph G is *semi-connected* if, for every pair of vertices u and v, either u is reachable from v or v is reachable from u (or both).

   (a) (5 points) Give an example of a DAG with a unique source (a source is a vertex with no entering edges) that is **not** semi-connected.

   (b) (10 points) Describe and analyze an algorithm to determine whether a given DAG is semi-connected.

   (c) (10 points) Describe and analyze an algorithm to determine whether an arbitrary directed graph is semi-connected.

4. An *Euler tour* of a strongly connected, directed graph $G = (V, E)$ is a cycle that traverses each *edge* of G exactly once, although it may visit a vertex more than once.

   (a) (10 points) Show that G has an Euler tour if and only if in-degree$(v) =$ out-degree(v) for each vertex $v \in V$.

   (b) (10 points) Describe an $O(|E|)$-time algorithm to find an Euler tour of G if one exists.