CS 170 Homework 3

Due **Monday 2/13/2023, at 10:00 pm (grace period until 11:59pm)**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write "none".

2 Finding Clusters

We are given a directed graph $G = (V, E)$, where $V = \{1, \ldots, n\}$, i.e. the vertices are integers in the range 1 to n. For every vertex i we would like to compute the value $f(i)$ defined as follows: $f(i)$ is the smallest j from which you can reach vertex i. (As a convention, we assume that i is reachable from i.)

(a) Show that the values $f(1), \ldots, f(n)$ can be computed in $O(|V| + |E|)$ time.

Please give a 3-part solution to this problem and write your algorithm in terms of pseudocode.

(b) Suppose we instead define $f(i)$ to be the smallest j that can be reached from i, instead of the smallest j from which you can reach i. How should you modify your answer to part (a) to work in this case?

3 Connectivity vs Strong Connectivity

(a) Prove that in any connected undirected graph $G = (V, E)$ there is a vertex $v \in V$ such that removing v from G gives another connected graph.

(b) Give an example of a strongly connected directed graph $G = (V, E)$ such that, for *every* $v \in V$, removing v from G gives a directed graph that is not strongly connected.

(c) Let $G = (V, E)$ be a connected undirected graph such that G remains connected after removing any vertex. Show that for every pair of vertices u, v where $(u, v) \notin E$ there exist at least two different u-v paths.

4 Semiconnected DAG

A directed acyclic graph G is *semiconnected* if for any two vertices A and B, there is either a path from A to B or a path from B to A. Show that G is semiconnected if and only if there is a directed path that visits all of the vertices of G.

5 Shortest Path Between Sets

Given a undirected weighted graph G with non-negative edge weights, let $d(s,t)$ be the shortest path length from s to t. Give an efficient algorithm that takes as input two subsets of vertices S and T and outputs $\min_{s \in S, t \in T} d(s,t)$, i.e. the shortest path from any vertex in S to any vertex in T. Provide a three-part solution.

(You may use a standard DFS/BFS/Dijkstra's algorithm as a subroutine in your solution.)

6 Coding Question

For this week's homework, we'll be going over some standard graph algorithms the a python jupyter notebook called `paths_on_graphs.ipynb`. There are two ways you can access the notebook and complete the problems:

1. Click here and navigate to the `HW3` folder if you prefer to complete this question on Berkeley DataHub.
2. Run

   ```
   git clone https://github.com/Berkeley-CS170/cs170-coding-notebooks-sp23
   ```

   in your computer's terminal (and navigate to the `HW3` folder) if you prefer to complete it locally. If you run into any issues with python import or autograder errors, please refer to the local setup instructions here.

Notes:

- *Submission Instructions:* Please download your completed `paths_on_graphs.ipynb` file and submit it to the gradescope assignment titled "Homework 3 Coding Portion".
- *OH/HWP Instructions:* There will be designated office hours for you to come to ask for conceptual and debugging help. Please visit the coding OH post for more information.
- *Academic Honesty Guideline:* We realize that code for some of the algorithms we ask you to implement may be readily available online, but we strongly encourage you to not directly copy code from these sources. Instead, try to refer to the resources mentioned in the notebook and come up with code yourself. That being said, we **do acknowledge** that there may not be many different ways to code up particular algorithms and that your solution may be similar to other solutions available online.