COMP 3711 – Design and Analysis of Algorithms
2024 Fall Semester – Written Assignment 4
Distributed: 9:00 on November 16, 2024
Due: 23:59 on November 29, 2024

Your solution should contain
(i) your name, (ii) your student ID #, and (iii) your email address
at the top of its first page.

Some Notes:

- Please write clearly and briefly. In particular, your solutions should be written or printed on *clean* white paper with no watermarks, i.e., student society paper is not allowed.

- Please also follow the guidelines on doing your own work and avoiding plagiarism as described on the class home page. ***You must acknowledge individuals who assisted you, or sources where you found solutions.*** Failure to do so will be considered plagiarism.

- The term *Documented Pseudocode* means that your pseudocode must contain documentation, i.e., comments, inside the pseudocode, briefly explaining what each part does.

- Many questions ask you to explain things, e.g., what an algorithm is doing, why it is correct, etc. To receive full points, the explanation must also be *understandable* as well as correct.

- Submit a SOFTCOPY of your assignment to Canvas by the deadline. If your submission is a scan of a handwritten solution, make sure that it is of high enough resolution to be easily read. At least 300dpi and possibly denser.

1. (20 points)

   (a) (5 points) Prove that if G is an undirected graph with n vertices and n edges with no vertices of degree 0 or 1, then the degree of every vertex is 2.

   (b) (5 points) Let G be an undirected graph with at least two vertices. Prove that it is impossible for every vertex of G to have a different degree.

   (c) (10 points) In a group of 10 people, each one has 7 friends among the other nine people. Prove that there exist 4 people who are friends of each other.

2. (20 points) Let $G = (V, E)$ be an undirected connected graph. Let n be the number of vertices in G. Let m be the number of edges in G. Design an algorithm to output a set of cycles $C_1, C_2, C_3, \ldots$ in G such that for every edge e of G, if e is contained in some cycle in G, then e is contained in some output cycle C_i. Explain the correctness of your algorithms. Analyze its running time which should be polynomial in n and m. Note that you are not required to output all cycles in G, and an edge of G may appear in multiple output cycles.

3. (20 points) Let $G = (V, E)$ be an undirected connected graph with n vertices and m edges. Each edge in G is also given an non-negative integer weight. Given a path P in G from a vertex u to a vertex v, the *bottleneck weight* of P, denoted by $wt(P)$, is the minimum edge weight in P. A *maximum bottleneck path* between u and v is the path Q between u and v such that $wt(Q) \geq wt(P)$ for all paths between u and v. Our problem is to report the maximum bottleneck paths between all pairs of vertices in G. Show that this problem can be solved by finding the minimum spanning tree of some graph. Explain the running time of your algorithm.

4. (20 points) Let $G = (V, E)$ be a directed graph with positive edge weights.

   (a) (10 points) The cost of a cycle is the sum of the weights of edges on that cycle. A cycle is called shortest if its cost is the minimum possible, Design an algorithm to return the cost of the shortest cycle in G. If G is acyclic, your algorithm should say so. Your algorithm should run in $O(n^3)$ time, where n is the number of vertices in G. Explain the correctness of your algorithm. Derive its running time.

   (b) (10 points) Suppose that the edge weights in G are integers from the given range $[0, W]$. Describe an implementation of Dijkstra's algorithm that runs in $O((n + m) \log W)$ time.

5. (20 points) Find the maximum flow from s to t in the following flow network. Determine the corresponding minimum cut as well. Follow the notation in the lecture slides to show your intermediate steps.