Algorithm Design and Analysis
Assignment 3

1. The following algorithm attempts to find the shortest path from node s to node t in a directed graph with some negative edges:

   - Add a large enough number to each edge weight so that all the weights become positive, then run Dijkstra's algorithm.

   Either prove this algorithm correct, or give a counter-example.

2. A bipartite graph is a graph $G = (V, E)$ whose vertices can be partitioned into two sets V_1 and V_2 (i.e., $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$) such that there are no edges between vertices in the same set. (For instance, if $u, v \in V_2$, then there is no edge between u and v.)

   (a) Show that an undirected graph is bipartite if and only if it contains no cycles of odd length.

   (b) Give a linear-time algorithm to determine whether an undirected graph is bipartite.

3. Let $G = (V, E)$ be an undirected connected graph. Let T be a depth-first search tree of G. Suppose that we orient the edges of G as follows: For each tree edge, the direction is from the parent to the child; for every non-tree (back) edge, the direction is from the descendent to the ancestor. Let G' denote the resulting directed graph.

   (a) Give an example to show that G' is not strongly connected.

   (b) Prove that if G' is strongly connected, then G satisfies the property that removing any single edge from G will still give a connected graph.

   (c) Prove that if G satisfies the property that removing any single edge from G will still give a connected graph, then G' must be strongly connected

   (d) Give an efficient algorithm to find all edges in a given undirected graph such that removing any one of them will make the graph no longer connected.

4. How long does it take you to finish the assignment (include thinking and discussing)? Give a score (1,2,3,4,5) to the difficulty.