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.

   Solution. It is not correct. Assume the number is c, the orginal weight of a path is w, then the new weight becomes $w' = w + kc$, where k is the length of the path.

   In a word, different paths get different additional weights.

   A counter-example: there are three vertices a, b, and c, where $w(a, b) = w(a, c) = 2, w(b, c) = -1$. The shortest path from a to c is $(a, b), (b, c)$ with weight 1, but it is replaced by (a, c) with weight $2 + c$ for any additional integer $c > 1$ in the modified graph. □

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.

   Solution. (b) DFS to color the vertices with two colors, where adjacent vertices must be colored differently. If conflict occurs, then this graph is not a bipartite graph. □

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.

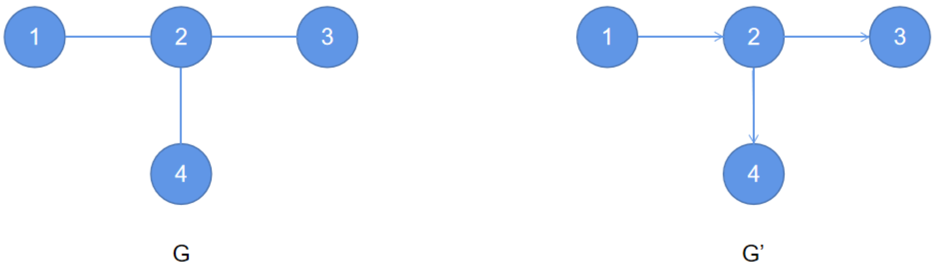
   (a) Give an example to show that G' is not strongly connected.

   (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.

Solution.



(a)

(b) Suppose that we remove (i, j) from G, and $(i, j) \in G', (j, i) \notin G'$ without loss of generality. If G' is strongly connected, there exists some path p from j to i in G'. So, the corresponding path in $G \backslash (i, j)$ makes i and j still connected, which implies that $G \backslash (i, j)$ is still connected.

(c) If we prove that both paths form i to j and j to i exists for arbitrary $(i, j) \in G$, G' is strongly connected.

Suppose that we remove (i, j) from G, and $(i, j) \in G', (j, i) \notin G'$ without loss of generality. If $G \backslash (i, j)$ is still connected, there exists some path p connect i and j in G. Assume it is $p_0, p_1, \ldots, p_k$ with length k where $p_0 = j$ and $p_k = i$.

Now prove that there exists some path p' form j to i in G'.

If the direction of (p_0, p_1) in G' is $j \to p_1$, then p_1 is a child node of j in the DFS tree, we have $(j, p_1) \in G'$. Otherwise, both i and p_1 pointing to j implies that p_1, i is a descendent of j.

1. If i is the descendent, (i, j) is the back edge, there is still a path form j(ancestor) to i(descendent). Now we get the proof.
2. If p_1 is the descendent, we replace (p_1, j) by a path from j to p_1 and denote it by $[j \to p_1]$

Whatever the direction of (p_1, j) is, we get a path from j to p_1. Applying this method on $(p_1, p_2), \ldots, (p_{k-1}, i)$, we get the path form j to i.

Since $G \backslash (i, j)$ is connected for any$(i, j) \in G$, $\forall p \neq q \in G$ there exists a path connecting p, q. For any edge in this path, the two vertices are reachable from each other, so p, q is also reachable from each other. G' is strongly connected.

Hint: Notice that a path in an undirected graph does not represent the same path in the corresponding directed graph, and a cut edge in a directed graph does not correspond to a cut edge in the undirected graph, either (Although they may hold in some special cases, you need to describe your proof in detail).

(d) Tarjan algorithm to find all of the cut edges.

□

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.