CS170 Discussion Section 4: *2/8*

1. *Short answer.* For each of the following, either prove the statement is true or give a counterexample to show it is false.

   (a) If (u, v) is an edge in an undirected graph and during DFS, $\text{post}(v) < \text{post}(u)$, then u is an ancestor of v in the DFS tree.

   (b) In a directed graph, if there is a path from u to v and $\text{pre}(u) < \text{pre}(v)$ then u is an ancestor of v in the DFS tree.

   (c) In any connected undirected graph G there is a vertex whose removal leaves G connected.

2. *Alien alphabet.*
   Suppose you have a dictionary of an alien language which lists words in some sorted lexicographical ordering. For example, given the following list of words:

   $$[\texttt{baa} \quad \texttt{abcd} \quad \texttt{abca} \quad \texttt{cab} \quad \texttt{cad}]$$

   You can conclude the ordering of the alphabet is

   $$\texttt{b} < \texttt{d} < \texttt{a} < \texttt{c}$$

   Give an efficient algorithm to determine the lexicographical ordering for any input list.

3. *True Source.* Design an efficient algorithm that given a directed graph G determines whether there is a single vertex v from which every other vertex can be reached. Hint: first solve this for directed acyclic graphs. Note that running DFS from every single vertex is not efficient.

4. *Shortest Cycle.*
   Give an algorithm that takes as input an undirected, unweighted graph, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^2 + |V| \cdot |E|)$.

5. *Dijkstra and negative edge weights.*
   In general, Dijkstra's algorithm doesn't work on graphs with negative edge weights. Here is one attempt to fix it:

   (a) Add a large number M to every edge so that there are no negative weights left.

   (b) Run Dijkstra to find the shortest path in the new graph.

   (c) Return the path Dijkstra's algorithm found, but with the old edge weights (subtract M from the weight of each edge).

   Show that this algorithm doesn't work — even when the graph is guaranteed not to contain a negative weight cycle — by finding such a graph for which the algorithm must give the wrong answer. What is your intuition for why this algorithm does not work?