1. Explain how a vertex $u$ can end up in a depth-first tree containing only $u$, even though $u$ has both incoming and outgoing edges in $G$.

2. Let $G$ be an undirected graph. Show how to use DFS to answer the following query in constant time: Are $u$ and $v$ connected?

3. Describe an algorithm that decides in linear time if a given graph is bipartite or not and output a bipartition in case it is.

4. Describe an algorithm that computes the length of a shortest odd cycle of a given graph $G$.

5. There is a new alien language which uses the Latin alphabet. However, the order among letters is unknown to you. You receive a list of non-empty words from the dictionary, where words are sorted lexicographically by the rules of this new language. Derive the order of letters in this language.

6. Prove or disprove: If a directed graph $G$ contains cycles, then TOPOLOGICAL-SORT produces a vertex ordering that minimizes the number of “bad” edges that are inconsistent with the ordering produced.

7. Another way to perform topological sorting on a directed acyclic graph $G = (V, E)$ is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Explain how to implement this idea so that it runs in time $O(|V| + |E|)$. What happens to this algorithm if $G$ has cycles?

8. How can the number of strongly connected components of a graph change if a new edge is added?

9. An Euler tour of a connected, directed graph $G = (V, E)$ is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once.

   (a) Show that $G$ has an Euler tour if and only if in-degree($v$) = out-degree($v$) for each vertex $v \in V$.

   (b) Describe an $O(|E|)$-time algorithm to find an Euler tour of $G$ if one exists.

10. Let $G = (V, E)$ be a directed graph in which each vertex $u \in V$ is labeled with a unique integer $L(u) \in \{1, 2, \ldots, |V|\}$. For each vertex $u \in V$, let $R(u)$ be the set of vertices reachable from $u$. Define min($u$) to be the vertex in $R(u)$ whose label is minimum. Give an $O(|V| + |E|)$-time algorithm that computes min($u$) for all $u \in V$. 
11. **(Cut vertices and biconnected components)** Let $G$ be an (undirected) graph. A cut vertex of $G$ is a vertex $u$ such that $G \setminus \{u\}$ has more connected components than $G$. A biconnected component of a graph $G$ is its maximal connected subgraph of $G$ with no cut vertex.

(a) Can biconnected components of a graph intersect?

(b) Explain how to modify the depth-first search to compute the following information: the depth of each vertex in the DFS tree, and for each vertex $v$, the lowest depth of neighbors of all descendants of $v$ (including $v$ itself) in the depth-first-search tree, called the lowpoint.

(c) Show that a non-root vertex $v$ is an articulation point iff there is a child $y$ of $v$ such that $\text{lowpoint}(y) \geq \text{depth}(v)$.

(d) Show that the root is a cut vertex if and only if it has at least two children.

(e) Show how to use the information about the cut vertices to compute the biconnected components.