Algorithms and Data Structures
Winter Term 2015/16
Exercises for Unit 20

1. Prove that the implementation for the DFS-framework presented in the lecture correctly identifies the strongly connected components in a directed graph. Convince yourself that similar arguments can also be applied to show correctness of the variant that computes 2-edge connected components in an undirected graph.

2. A *directed acyclic graph (DAG)* is a directed graph $G = (V, E)$ without a directed cycle.
   a) Prove that a DAG has a topological ordering, i.e., an assignment $\pi : V \to \mathbb{N}$ such that $\pi(v) > \pi(w)$ for every edge $(v, w) \in E$.
   b) Design a DFS-algorithm that computes in $O(n + m)$ time a topological ordering if G is a DAG, or outputs a directed cycle if G is not a DAG.

3. An Euler tour of a (strongly) connected directed graph $G = (V, E)$ is a cycle that traverses every edge of G exactly once, but 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 all $v \in V$.
   b) Describe an $O(m)$-time algorithm to find an Euler tour of G if one exists.