Algorithms: Assignment 8

Due date: November 9 (Thursday)

Problem 1 (4 points)
The depth-first search algorithm may be used to identify the connected components of an *undirected* graph. Write a modified version of DFS for this task.

Your algorithm must determine the number k of connected components in an undirected graph and return this number. Furthermore, for every vertex u, the algorithm must assign an integer label $component[u]$, between 1 and k, that denotes the corresponding connected component. If two vertices are in the same component, they must get the same label. On the other hand, if vertices are in different components, their labels must be distinct.

Problem 2 (6 points)
Suppose that G is an *undirected* graph, and you need to check whether G has cycles. Design an algorithm $\textsc{Acyclic}(G)$ that returns TRUE if G is acyclic, and FALSE if G has cycles. Its running time should be $O(V)$, regardless of the number of edges.

Problem 3 (bonus)
This problem is optional, and it allows you to get 2 bonus points toward your final grade.

Consider a small country that consists of five towns, located in the vertices of a regular pentagon. The king has decided to connect the towns by roads, and his plan is to build a road from every town to every other town. The roads can curve, but they cannot intersect each other, and builders cannot use bridges. Can the builders satisfy the king's requirements? If yes, draw an appropriate layout of roads; if no, *prove* that such a layout does not exist.