George Washington University
Department of Computer Science

Csci 6212 - Homework 9

Given: April 19, 2017
Due: 6pm, April 25, 2017

1. The input is a connected undirected graph G, a spanning tree T of G, and a vertex $v \in V(G)$. Design an $O(|E|)$ time algorithm to determine whether T is a valid DFS tree of G rooted at v, i.e., determine whether T can be an output of DFS under some order of edges starting with v.

2. Characterize conditions of undirected graphs that contain a vertex v such that there exists a DFS tree rooted at v that is identical to a BFS tree rooted at v. Note that two spanning trees are identical if they contain the same set of edges.

3. Given a undirected graph G, the length (no weight) of the smallest cycle is called the *girth* of the graph. Design an $O(|E|)$ time algorithm to compute the girth of a given graph G.

4. Given a connected undirected graph G and three edges (u_1, v_1), (u_2, v_2), (u_3, v_3) of $E(G)$, design an $O(|E|)$ time algorithm to determine whether there exists a cycle in G that contains both (u_1, v_1) and (u_2, v_2), but does not contain (u_3, v_3).

5. Depth-first numbers are given to the vertices in the graph G shown in Figure 1.

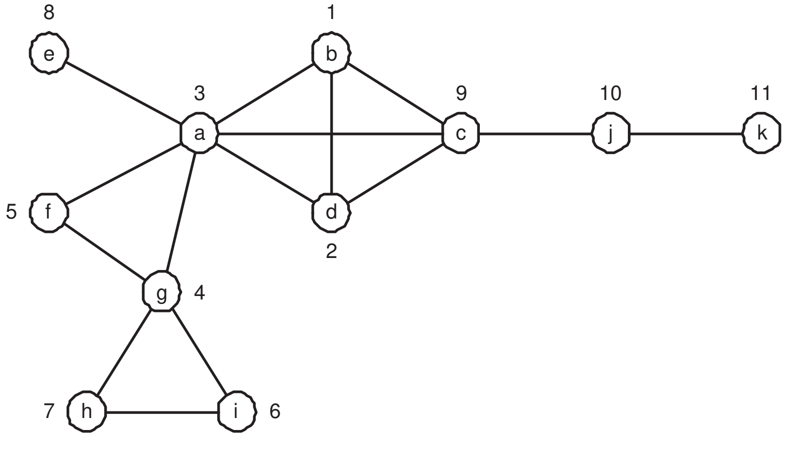

Figure 1:

(a) Show the depth-first-search tree of G corresponding to the depth-first numbers.

(b) Compute the *low point* of each node in G using the definition of the low point (not by an algorithm).

6. Consider the following problem : *Is there a vertex v in G such that every other vertex in G can be reached by a path from v ?* If G is an undirected graph, the question can be easily answered by a simple depth-first (or breadth-first) search and a check to see if every vertex were visited. Describe an algorithm to solve for a directed graph. What is the complexity of your algorithm?

7. Consider a graph G.

(a) Explain how one can check a graph's acyclicity by using breadth-first-search.

(b) Foes either of the two search algorithms, DFS and BFS, always find a cycle faster than the other? If your answer is yes, indicate which of them is better and explain why it is the case; if your answer is no, give two examples supporting your answer.

8. A graph is called *bipartite* if all its vertices can be partitioned in two two disjoint subsets $X$and Y so that every edge connects a vertex in X with a vertex in Y.

(a) Design a DFS-based algorithm for checking whether a graph is bipartite.

(b) Design a BFS-based algorithm for checking whether a graph is bipartite.