Algorithms

Assignment: Graph Traversals

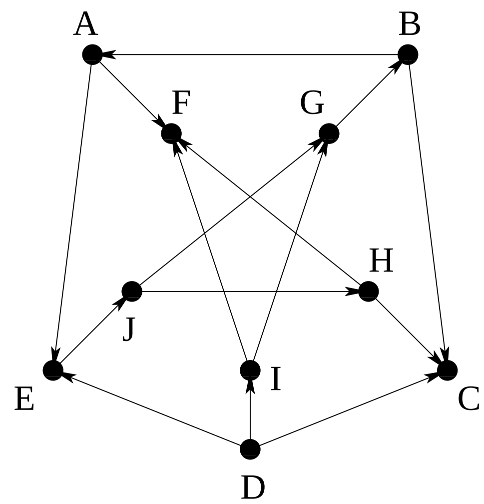
Name: ....

Id: ....

Grade

Good Luck!

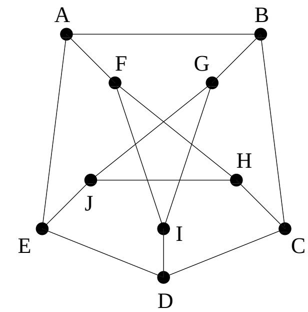
1.



(a) Run the DFS algorithm on the above directed Petersen graph.
Start with A and assume that the vertices are ordered: $A, B, C, D, E, F, G, H, I, J$.
Write the time intervals for each of the 10 vertices.

(b) Draw the DFS forest and classify the edges as tree, forward, back, or cross edges.

2.



(a) Draw the BFS tree for the above undirected Petersen graph.
Start with A and assume that the vertices are ordered: $A, B, C, D, E, F, G, H, I, J$.
Draw all the edges and write the level of each of the 10 vertices.

(b) Draw the BFS tree for the above undirected Petersen graph.
Start with J and assume that the vertices are ordered: $J, I, H, G, F, E, D, C, B, A$.
Draw all the edges and write the level of each of the 10 vertices.

(c) The Petersen graph is very symmetric. Support this claim with your trees from the previous two parts.

3. Let $G = (V, E)$ be a simple undirected connected graph with n vertices.

The BFS and the DFS traversals generate rooted and ordered traversal trees. The structure and the height of these trees may depend on the starting vertex and the order among the vertices when applying the traversals procedures.

The height of a rooted tree is the distance in edges from the root to the furthest leaf. For example, the height of a tree with one vertex is 0, the height of a path tree for which the root is one of its end vertices is $n - 1$, and the height of a star tree for which the center vertex is the root is 1.

For each one of the following cases determine the possible structure of the traversal tree and its possible maximum and minimum height. You may assume that n is large enough.

Graph	Traversal	structure(s)	max. height	min. height
Complete	DFS			
Complete	BFS			
Path	DFS			
Path	BFS			
Cycle	DFS			
Cycle	BFS			
Star	DFS			
Star	BFS			

Illustrate all possible structures of the traversals trees generated by DFS on the wheel graph W_n.

Illustrate all possible structures of the traversals trees generated by BFS on the wheel graph W_n.

4. Let G be a directed graph represented by the adjacency matrix A. In directed graphs: $A(u, v) = 1$ if $(u \to v)$ exists and $A(u, v) = 0$ otherwise. A *tournament* is a directed graph such that for any pair $u \neq v$ exactly one of $(u \to v)$ or $(v \to u)$ exists. An *acyclic* tournament is a tournament with no cycles. A *source* is a vertex whose in-degree is zero. A *destination* is a vertex whose out-degree is zero.

   (a) Describe an algorithm that checks if a directed graph G is a tournament. What is the complexity of your algorithm? Explain.

(b) Prove that in an acyclic tournament there is exactly one source and one destination. Describe an efficient algorithm ($O(n)$-time) to find the source vertex of an acyclic tournament.