Module 2 - Graphs: Basic Written HW

1. Let G be an undirected graph with n nodes (let's assume n is even). Prove or provide a counterexample for the following claim: If every node of G has a degree of at least $\frac{n}{2}$, then G must be connected.

2. Most graph algorithms that take an adjacency-matrix representation as input require time $\Omega(V^2)$, but there are some exceptions. Show how to determine whether a directed graph G contains a universal sinka vertex with in-degree $|V| - 1$ and out-degree 0 in time $O(V)$, given an adjacency matrix for G.

3. The textbook describes two variables that can be associated with each node in a graph G during the execution of *Depth-First Search*: discovery time ($v.d$) and finish time ($v.f$). These are integer values that are unique. Every time a node is discovered (i.e., DFS sees the node for the first time) that node's $v.d$ is set to the next available integer. When DFS is finished exploring ALL of this node's children, $v.f$ is set to the next available integer.

   For this question, consider a single edge in a graph G after DFS finishes executing. You might need to reference the textbook or slides for definitions of tree edge, forward edge, back edge, and cross edge. Argue that each edge $e = (u, v)$ is:

   1. A tree edge or forward edge if and only if $u.d < v.d < v.f < u.f$
   2. A back edge if and only if $v.d \le u.d < u.f \le v.f$
   3. A cross edge if and only if $v.d < v.f < u.d < u.f$

   You can describe your answers intuitively, but your answers must be clearly articulated.

4. *Kruskal's algorithm* begins by adding the smallest edge in the graph to the solution (and never looking back). Let $e = (u, v)$ be a minimum-weight edge in a connected graph G. Show that $e = (u, v)$ belongs to some minimum spanning tree of G. *HINT: Use a proof by contradiction. Note that e will eventually connect two smaller spanning trees together. If e is NOT in the solution, than something else IS. Show that this leads to some kind of contradiction.*