CSE 5311.004 Fall 2005

Exercise Set Weeks 6, 7 and 8

1. Design an algorithm to check whether an undirected graph G = (V, E) has an Euler cycle (find out what Euler cycle means). What is the complexity of your algorithm?
2. Consider a directed graph G = (V, E) that has no cycles. Design an efficient algorithm that outputs the vertices in a sequence $\{v_1, v_2 \ldots v_n\}$ such that for all $i < j$, there is no edge from v_j to v_i. (Hint: use depth-first search)
3. Suppose you are given a graph that is "almost disconnected", i.e., the removal of a single edge will disconnect the graph. Design an algorithm to efficiently find this edge.
4. What are the kinds of graphs for which depth-first search orderings and breadth-first search orderings are the same?
5. Analyze the running time of Kruskal's minimum spanning tree algorithm when the input is an adjacency matrix.
6. Design an example of a graph where the shortest path tree is longer than the minimum spanning tree. In the worst case, how much longer can the shortest path tree be than the minimum spanning tree?
7. A "geometric graph" is a special type of graph where the nodes are points on a 2-dimensional surface and edges are straight lines joining pairs of nodes.
    a. Show that the minimum spanning tree of such graphs cannot have edges that cross each other (other than at their endpoints).
    b. Show that the same property holds for shortest path trees.
8. Incremental tree calculations:
    a. Suppose you have constructed a minimum spanning tree of a graph. A new edge is now inserted into the graph. Design an algorithm that will efficiently compute the new MST.
    b. Suppose you have constructed a shortest path tree from a given vertex of a graph. A new edge is now inserted into the graph. Design an algorithm that will efficiently construct the new shortest path tree.