Homework #3: Elementary Graph Algorithms, Minimum Spanning Tree
Due Date: Tuesday, 11 March 2003

Guidelines

Please make sure you adhere to the policies on collaboration and academic honesty as outlined in Handout #1.

Reading

Read Ch. 22, 23 of CLRS.

Practice

These exercises are purely for your own practice. You should not turn them in, and you are free to discuss them fully with others.

- Do CLRS 22.2-1
- Do CLRS 22.3-2
- Do CLRS 22.3-4
- Do CLRS 22.3-8
- Do CLRS 22.3-10
- Do CLRS 22.4-1

Problems

Problem A (36 points) "Work entirely on your own."

In this problem, we wish to study the inner-workings of the <u>depth-first</u> search algorithm on a <u>directed</u> graph. One issue we wish to explore is the various types of edge classifications that arise (tree/back/forward/cross). The other issue we wish to explore is the coloring of vertices as the algorithm is in progress (WHITE/GRAY/BLACK).

When a DFS begins, all vertices are initialized to WHITE. By the end of a search, all vertices are BLACK. However, as the search progresses, you will find edges which connect vertices of varying colors.

Please fill in a chart formatted as follows:

		"from"		
		WHITE	GRAY	BLACK
	WHITE			
"to"	GRAY			
	BLACK			

For each such cell $(i.j)$, indicate whether, at any point during a depth-first search of a directed graph, there might be an edge from a vertex of color i to a vertex of color j.

Furthermore, for those combinations where edges might exist, indicate what types of edges are possible.

You do not need to provide any justification of your (correct) answers.

Problem B (15 points) "Work entirely on your own."

Give a counterexample to the conjecture that if there is a path from u to v in a directed graph G, and if $d[u] < d[v]$ in a depth-first search of G, then v is a descendant of u in the depth-first forest produced.

Problem C (15 points) "You may discuss ideas with other students."

Prove or disprove: If a directed graph G contains cycles, then a call to Topological-Sort(G) of Chapter 22.4 produces a vertex ordering that mnimizes the number of "bad" edges that are inconsistent with the ordering produced.

Problem D (20 points) "You may discuss ideas with other students."

Another way to perform topological sorting on a directed acyclic graph $G = (V, E)$ is the following:

while vertices remain:
  Find a vertex v of in-degree 0.
  Output v as the next item in the topological order.
  Remove v and all of its outgoing edges from the graph.

That the resulting order is indeed a valid topological sort of the graph is relatively easy to show; you are not asked to give such a proof.

Your goal is to explain how to implement this idea so that it runs in time $O(V + E)$.

Note: you do not necessarily need to give code, but you should definately give pseudocode including a detailed explanation of exactly what data structures you use, and a justification of the claimed running time.

Problem E (14 points) "You may discuss ideas with other students."

A "light" edge crossing a cut is defined on page 563 of CLRS.

If, for every cut of a graph there is a unique "light" edge crossing the cut, show that this graph has a unique minimum spanning tree.

Show that the converse is not true. That is, provide an example of a graph which has a unique minimum spanning tree, even though there exists a cut of that graph for which there is no unique "light" edge.

Problem F **(EXTRA CREDIT – 10 points)**

"You may discuss ideas with other students."

Let T be a minimum spanning tree of graph G, and let L be the sorted list of the edge weights of T. Show that for any other minimum spanning tree T' of G, the list L is also the sorted list of edge weights of T'.