CSE 373 Homework 4 Sample Solution

June 7, 1998

(note that the last three problems were dropped from the homework, so only exercises 9.1, 9.5 and 9.7a were required)

1. (9.1) To find a topological ordering of a graph we find a vertex with no in edges, append it to the topological ordering, and remove it from the graph. Repeat this process until no vertices remain. For the given graph there are several solutions, one of which is $sGDABHEIFCt$.

2. (9.5a) To find the shortest weighted path from A to all other vertices we apply Dijkstra's algorithm. Begin by setting Weight(V) to ∞ and Predecessor(V) to NULL for all vertices V other than A. Next set Predecessor(B) and Predecessor(C) to A, Weight(B) to 5, Weight(C) to 3, and remove A from the graph. C has the minimum Weight so we consider it next, setting Predecessor(D) and Predecessor(E) to C, Weight(D) and Weight(E) to 10 ($3+7$), and then removing C from the graph. Now Weight(B) is minimum, so we set Predecessor(G) and Predecessor(E) to B, Weight(D) to 6, Weight(E) to 8, and remove B from the graph (we only change E because $8 < 10$). We repeat this process until all vertices have been considered, then find the shortest paths by tracing the Predecessors. This yields the following table:

path from A to:	path weight:	path:
B	5	$A \to B$
C	3	$A \to C$
D	9	$A \to B \to G \to E \to D$
E	7	$A \to B \to G \to E$
F	8	$A \to B \to G \to E \to F$
G	6	$A \to B \to G$

3. (9.5b) To find the shortest unweighted path from B to all other vertices we apply essentially the same algorithm, except we consider all edge weights to be 1. This yields the following table:

path from B to:	path length:	path:
A	3	$B \to C \to D \to A$
C	1	$B \to C$
D	2	$B \to C \to D$
E	1	$B \to E$
F	2	$B \to E \to F$
G	1	$B \to G$

4. (9.7a) Consider the graph with vertex set A, B, C, an edge from A to B of weight 2, an edge from A to C of weight 3, and an edge from C to B of weight -2. Let us use Dijkstra's algorithm to calculate the shortest path from A to all other vertices. We begin by initializing Weight(V) to ∞ and Predecessor(V) to NULL for $V \in B, C$. Next we set Predecessor(B) and Predecessor(C) to A, Weight(B) to 2, Weight(C) to 3, and remove A from the graph. Weight(B) is minimum, so we consider B next. B has no out edges, we remove it from the graph. C is the only remaining vertex, so we're done. This yields the table

path from A to:	path weight:	path:
B	2	$A \to B$
C	3	$A \to C$

when the actual shortest weighted path from A to B is given by

path from A to:	path weight:	path:
B	1	$A \to C \to B$
C	3	$A \to C$

Thus, Dijkstra's algorithm isn't guaranteed to work in the presence of negative weight edges. However, it will always work provided all edges have weight at least 0.