CS140: Algorithms

Fall 2000

Assignment 11

This assignment is due by the start of class on Wed., Nov. 6. You may collaborate with your classmates to complete this assignment but you must reference your collaborators. In addition, you must do your own write-up. This assignment is worth 100 points (50 points if handwritten). You may, however, submit a handwritten solution to the network flow problem.

Your answers will be judged for correctness, clarity, and conciseness. When you give a recursive definition or algorithm, you should begin with a brief (a few sentences) overview that identifies the key ideas behind your approach. Then fill in the (important) details.

1. The Maximum Weight Spanning Tree problem, as its name implies, asks for a maximum weight spanning tree of the (connected) input graph. Show how to reduce the Maximum Weight Spanning Tree problem to MST.

2. A Bottleneck Spanning Tree (BST) of a connected graph G is a spanning tree whose maximum weight edge is a small as possible. In other words, the cost of a spanning tree for this problem is the weight of its maximum weight edge. The problem is to find a spanning tree with minimum cost. (Assume G has n vertices and m edges.)

   (a) Suppose T is an MST G. Show that T is also a BST of G.

   (b) Give a weighted, connected graph G and a BST, T, of G such that T is not an MST of G.

   (c) By part (a) we can clearly reduce BST to MST. But by part (b) we might guess that there is a more efficient algorithm for finding a BST of G.

      i. Suppose I tell you that the cost of a BST of G is W. Give an $O(m)$ algorithm that then finds a BST of G.

      ii. Suppose I simply want to know if the weight of a BST of G is less than or equal to W. Give an $O(m)$ algorithm that answers this question.

      iii. Now design an $O(m)$ algorithm that finds a BST of G. Hint: Use divide and conquer.

3. Run the Floyd-Warshall algorithm on the digraph whose adjacency matrix is given below. Show D^k for $k = 0, \ldots, 5$. (There are no self-loops, hence the 0's on the diagonal. An ∞ indicates a lack of edge.)

0	∞	∞	-1	20
∞	0	5	3	∞
∞	-2	0	1	∞
1	2	2	0	∞
3	3	2	∞	0