CSC 373 Midterm 2, Winter 2010

Name: ___________________________

Student Id: ________________________

Please put your name on each page of this midterm.

Advice: The midterm has three questions, together worth 50 points. We expect you to spend as many minutes on a problem as it is worth points. You should read through the whole midterm, and start with the problem you find easiest. You will get more credit for doing one problem well then for doing two poorly. If you cannot answer a question, you will receive 10% of the marks for that question if you leave it blank. Good luck.

Name:

Problem 1 (25 points): Dynamic Programming
There are n trading posts along a river. At any of the posts you can rent a canoe to be returned at another post downstream. (It is impossible to paddle against the current.) For each possible departure point i ($1 \leq i \leq n$) and each possible arrival point j ($i \leq j \leq n$), the cost of a rental from i to j is given by p_{ij} . However, it can happen that the cost of renting from i to j is higher than the total cost of a series of shorter rentals. In this case, you can return the first canoe at some post k between i and j and continue your journey with a second canoe. There is no extra charge for changing canoes in this way. The goal is to determine the minimum cost to travel from post 1 to post n. You are asked to design a dynamic programming algorithm for this problem.

(a, 5 points) Give a DP recurrence that describes the minimum cost of commuting from the starting post to an arbitrary post j. Briefly explain your recurrence.

(b, 10 points) Write pseudocode for solving this problem

Name:

Your partner from the project suggests that instead of coding this, you can reduce the problem to the Shortest Path problem (where the posts are nodes, and edges between them have lengths equal to the cost of renting a canoe between those posts). He suggests using Dijkstra's algorithm, which has running time O(E log V), to find the path.

(c, 4 points) Is your partner right? That is, is the reduction valid? Explain why/why not.

(d, 6 points) Assume that the reduction in part i) is valid. Give the running time for the two algorithms in terms of n, and state which one is faster.

Name:

Problem 2 (18 points): Network Flow
The **Global Min-Cut** problem is a variation of the Min-Cut problem that we studied in connection with network flow. Our input is an **undirected** graph G with n vertices, m edges, and a positive integer weight on each edge. We want to find the smallest (lowest weight) possible cut (A,B) in G, where A and B are disjoint **non-empty** sets of vertices with A ∪ B = V (the set of all vertices). There are two differences between this problem and the Min-Cut problem we studied -- the graph is undirected, and there are no designated vertices S and T that must be on opposite sides of the cut. You are asked to develop an algorithm for this problem.

(a, 10 points) First we will solve the **Undirected Min-Cut** problem. You are given a graph with undirected edges, each with a capacity, and two vertices S and T. You are asked to compute the smallest cut that will separate S from T. Show a reduction from this problem to regular, directed, network flow. Explain exactly how you will build a directed network, and what you will do to the solution to the flow problem on this network to identify those edges of the input graph that are in the cut.

Name:

(b, 8 points) Now use the algorithm developed in part (a) to build an algorithm for the **Global Min-Cut** problem. You can use the Undirected Min-Cut problem as a subroutine even if you were unable to solve part (a).

Problem 3 (7 points): Simple proof.

If the capacities of all of the edges in a directed network are multiples of 3, then the max-flow will be a multiple of 3. Prove the statement or provide a counter-example.