University of Washington
Department of Computer Science and Engineering
CSE 421, Fall 2005

November 28, 2005

Practice Final Exam

NAME: ______________________________

Instructions:

- Closed book, closed notes, no calculators
- Time limit: 1 hour 50 minutes
- Answer the problems on the exam paper.
- If you need extra space use the back of a page
- Problems are not of equal difficulty, if you get stuck on a problem, move on.
- This practice exam is based on the CSE 521 exam from Spring 1998, some changes have been made to reflect different coverage and content.

1	/15
2	/10
3	/20
4	/10
5	/25
6	/10
7	/10
Total	/100

Problem 1 (15 points):

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} 3T(n/3) + n^2 & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases}$$

b)

$$T(n) = \begin{cases} 4T(n/3) + n & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases}$$

c)

$$T(n) = \begin{cases} T(n/3) + 1 & \text{if } n > 1 \\ 0 & \text{if } n = 0 \end{cases}$$

Problem 2 (10 points):

Let $G = (V, E)$ be an undirected, bipartite graph. A matching M is said to be *maximal* if every edge of E shares at least one endpoint with an edge of M. Let M_{opt} be a maximum cardinality matching for G.

a) Give an example of a graph G, a maximal matching M_{greed} where $|M_{greed}| = \frac{1}{2}|M_{opt}|$.

b) For a bipartite graph G, and maximal matching M_{greed}, prove that $|M_{greed}| \geq \frac{1}{2}|M_{opt}|$.

Problem 3 (20 points):

How can you:

a) Find a maximum weight spanning tree using an algorithm which computes a minimum weight spanning tree.

b) Find a shortest path in an undirected graph, using an algorithm which finds a shortest path in a directed graph.

c) Find a female optimal stable marriag, given an algorithm for a male optimal stable marriage.

d) Find a maximum flow in a graph with capacities on the vertices (meaning a bound on the flow that can go into each vertex), using an algorithm for maximum flow where the capacities are on the edges.

Problem 4 (10 points):

Let $G = (V, E)$ be a directed graph with edge costs, K an integer, and s and t vertices of V. Describe an algorithm which finds the most expensive path from s to t that uses exactly K edges.

Problem 5 (25 points):

What is the fastest known algorithm for each of the following problems? Give a short description or citation (no more than two sentences each). What is the run time of the fastest algorithm?

1. Determining if an undirected graph with n vertices and m edges is bipartite.

2. Computing the longest common subsequence of a pair of strings each of length n.

3. Solving the single source shortest paths problem on a graph with n vertices and m edges.

4. Solving the single source shortest paths problem on an *acyclic* graph with n vertices and m edges.

5. Checking whether or not an undirected graph with n vertices and m edges has a cycle.

6. Given n points in the plane, find the closest pair of points.

7. Solving the knapsack problem with n items, and a bound of K on the size of the knapsack.

8. Find a maximum cardinality matching in a bipartite graph with n vertices and m edges.

Problem 6 (10 points):

Give short answers to the following questions about network flow:

a) Is the Ford-Fulkerson algorithm a polynomial time algorithm? Why or why not.

b) How do you find the minimum cut of a graph, after a flow algorithm has found the maximum flow?

Problem 7 (10 points):

Suppose that you have an algorithm for determining if a graph with vertex demands and edge capacities has a feasible circulation. How would you use this algorithm to find the value of the maximum flow in a graph with edge capacities and a source and a sink.