University of Washington March 21, 2001
Department of Computer Science and Engineering
CSE 521, Spring 1998

Final Exam, June 8, 1998

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.

1	/15
2	/10
3	/25
4	/15
5	/30
6	/15
7	/10
Total	/120

Problem 1 (15 points):

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} 1 + \sum_{i=1}^{n-1} T(i) & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

b)

$$T(n) = \begin{cases} T(n-1) \cdot T(n-1) & \text{if } n > 0 \\ 2 & \text{if } n = 0 \end{cases}$$

c)

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

Problem 2 (10 points):

Let $G = (V, E)$ be an undirected 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, and M_{greed} be a maximal matching. Prove that $|M_{greed}| \geq \frac{1}{2} |M_{opt}|$.

Problem 3 (25 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 maximum independent set in a graph using an algorithm which finds a maximum clique in a graph.

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.

e) Find an optimal travelling salesman tour in a graph which may have negative length edges with an algorithm for the travelling salesman problem that requires all edges have positive length.

Problem 4 (15 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 cheapest path from s to t that uses exactly K edges.

Problem 5 (30 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. Computing the median of a set of n integers.

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 a graph with n vertices has a Hamiltonian circuit.

6. Computing the convex hull of a set of n points in the plane.

7. Computing the product of two $n \times n$ matrices.

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

9. Determining if there is a feasible solution to a set of linear inequality constraints.

10. Finding a travelling salesman tour within a factor of two of optimal (assuming that the edge costs satisfy the triangle inequality).

Problem 6 (15 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?

c) What are the two fundamental operations in Goldberg's push-relabel algorithm. Briefly describe what the operations do, and what the preconditions are for applying them.

Problem 7 (10 points):

Show that the following problems are NP complete (you do not need to show the problems are in NP):

a) 4SAT (Satisfiability with exactly four literals per clause). Give a reduction from 3SAT.

b) Two processor scheduling without precedence constraints. Each job j_i has an integer execution time t_i. Once a job is started, it must be run until completion. The question is to determine if the jobs can be executed on two processors, and all jobs complete by a given time T.