University of Washington
Department of Computer Science and Engineering
CSE 521, Winter 2000

March 16, 2000

Final Exam, March 16, 2000

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	/20
2	/20
3	/20
4	/20
5	/20
6	/20
7	/20
Total	/140

Problem 1 (20 points):

Give asymptotic solutions to the following recurrences. Justify your answers.

a)

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

b)

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

c)

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

d)

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

Problem 2 (20 points):

Give short answers to the following questions.

a) What is your favorite matroid?

b) Correctly spell the name of a well known Dutch computer scientist.

c) For the median problem, there is a linear expected time randomized algorithm, which is much simpler than the determinstic linear time algorithm. Why is the randomized algorithm so much simpler?

d) How does Strassen's algorithm achieve an asymptotic improvement in runtime over the $O(n^3)$ matrix multiplication algorithm?

e) How fast can one determine the convex hull of a set of n points in the plane.

f) Why is the triangle inequality important in finding approximate solutions to the Travelling Salesman Problem.

g) What is the maximum number of phases for the augmenting path algorithm for network flow (Ford-Fulkerson) when the edges have unit capacity.

h) How much space is required to find the Longest Common Subsequence of sequences S and T of lengths m and n respectively.

Problem 3 (20 points):

Given a directed graph G with edge costs $c(e)$ for $e \in C$, the minimum-mean cost cycle problem is to find a cycle C which minimizes: $\frac{1}{|C|}\sum_{e \in C}$.

a) Show that there is a solution where the minimum-mean cost cycle is a *simple* cycle. (I.e., no vertices are visited more than once).

b) Give a polynomial time algorithm to solve the minimum-mean cost cycle problem. (Important hint - first find the minimum cost cycle of length K).

Problem 4 (20 points):

The precedence constrained multiprocessor scheduling problem is to assign processors and time slots to jobs so that every job is scheduled after all of its predecessors.

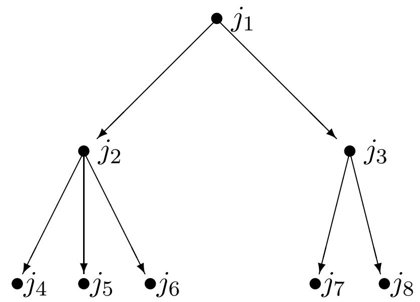
The input to the problem is a directed acyclic graph and an integer m, and the output should be a valid schedule for the jobs on m processors.

Assume that the jobs all have unit cost, and that the precedence graph is an out-tree with all leaves the same distance from the root.

Give a polynomial time algorithm to find an optimal schedule, and argue that your algorithm is correct.

As an example, consider the following precedence. An optimal, 2-processor schedule is:

$$(j_1, -), (j_2, j_3), (j_4, j_5), (j_6, j_7), (j_8, -)$$



Problem 5 (20 points):

Let $G = (V, E)$ be an undirected graph, with non-negative costs $c(e)$ assigned to the edges. Assume that no two edges have the same cost, so that the minimum spanning tree T is unique.

Suppose that we change the cost function to $c'(e) = 2^{c(e)} + c(e) + 37$. Prove that T is also the minimum spanning tree under c'.

Problem 6 (20 points):

Let $G = (V, E)$ be a network, and let f_1 and f_2 be valid flows on G. For $0 \leq \alpha \leq 1$, prove that $\alpha f_1 + (1 - \alpha) f_2$ is a valid flow on G.

Problem 7 (20 points):

Consider the following approximation algorithm for the knapsack problem, which constructs two candidate solutions and returns the one with highest value.

The first is formed by starting with the optimal fractional solution, and taking the items which are fully placed in the knapsack.

The second is formed by taking just the maximum value item which fits in the knapsack.

Show that this is a $\frac{1}{2}$ approximation algorithm for the knapsack problem.

Hint: the second candidate solution is needed to cover the case where the fractional solution has a large, high value item that doesn't fully fit. For example, if item 1 has size .2 and value 2, and item 2 has size .9 and value 8, and all other items have value 0, the fractional solution will take all of item 1 and $\frac{8}{9}$ of item 2.

The fractional solution only has one partial item, so consider the case where the partial item represents more than half the value, and the case where the partial item has less than half the value.

You may assume that each item has a size which is no larger than the knapsack.