University of Washington
Department of Computer Science and Engineering
CSE 421, Fall 2006

December 10, 2006

Final Exam, Monday, December 11, 2006

NAME: ____________________

Instructions:

- Closed book, closed notes, no calculators
- Time limit: One 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	/30
2	/30
3	/30
4	/30
5	/20
6	/20
7	/20
8	/20
9	/20
Total	/220

Problem 1 (30 points) Recurrences:

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} 5T(n/5) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

b)

$$T(n) = \begin{cases} 2T(n/5) + n^{1/2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

c)

$$T(n) = \begin{cases} 6T(n/5) + n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Problem 2 (30 points)Maximal Independent Set:

Let $G = (V, E)$ be an undirected graph. A subset I of the vertices is said to be independent if for all $u, v \in I$, $(u, v) \notin E$. A set of vertices M is a *maximal* independent set if M is not contained in any larger independent set, i.e., if $M \subset M'$ then M' is not independent. A set of vertices M is a *maximum* independent set if it is a largest independent set in the graph, i.e., M is independent, and if M' is any other independent set of G, $|M'| \leq |M|$. In other words, a maximal independent set is an independent set that we cannot add any more vertices to without it ceasing to be independent, a maximum independent set is an independent set that contains as many vertices as possible.

a) Show that a maximal independent set is not necessarily a maximum independent set.

b) Show that a maximum independent set is a maximal independent set.

c) Give a polynomial time algorithm that finds a maximal independent set in a graph.

Problem 3 (30 points) Short Answer:

a) Compute the Fourier Transform of $1 + 2x + x^2$ at four points.

b) Why does Dijkstra's algorithm not work for graphs with negative cost edges.

c) What is Cook's theorem?

d) If you know problem X is NP-complete, and you want to show that problem Y is NP-complete, do you reduce X to Y, or reduce Y to X. Justify your answer.

e) Name two minimum spanning tree algorithms. Which one could be called "Dijkstra's algorithm for minimum spanning trees".

f) If you have n points in the plane, what is the run time of the fastest known algorithm for finding the closest pair of points.

Problem 4 (30 points) Longest Path Problem:

For each of the following problems sketch an efficient algorithm or give an NP-completeness proof. You may draw on results from class or the textbook. (Note: In this problem, path length is the number of edges.)

a) Longest path in a directed acyclic graph. Given vertices s and t, find a maximum length path from s to t.

b) Path of length K. Given a directed graph, vertices s and t, and an integer K, find a path of length exactly K between s and t.

c) Longest simple path. Given a directed graph and vertices s and t, find the longest simple path between s and t. (Recall that a simple path is a path with no repeated vertices.)

Problem 5 (20 points) Non-adjacent LCS:

The sequence $C = c_1, \ldots, c_k$ is a *non-adjacent subsequence* of $A = a_1, \ldots, a_n$, if C can be formed by selecting non-adjacent elements of A, i.e., if $c_1 = a_{r_1}, c_2 = a_{r_2}, \ldots, c_k = a_{r_k}$, where $r_j < r_{j+1} - 1$. The non-adjacent LCS problem is given sequences A and B, find a maximum length sequence C which is a non-adjacent subsequence of both A and B.

This problem can be solved with dynamic programming. Give a recurrence that is the basis for a dynamic programming algorithm. You should also give the appropriate base cases, and explain why your recurrence is correct.

Problem 6 (20 points) Mod K Subset Sum:

The mod K subset sum problem is: given a set of integers $S = \{s_1, \ldots, s_n\}$ and an integer K, does there exist a non-empty subset S' of S such that $\sum_{s \in S'} s = W$ with $W \bmod K = 0$.

Give a dynamic programming algorithm for solving the mod K subset sum problmem. Your algorithm should return the set S' if it exists.

Problem 7 (20 points) Capacity Reduction for NetFlow:

Let $G = (V, E)$ be a flow graph with maximum flow f. Let e be an edge with capacity c. Suppose that the edge has it's capacity reduced to $c-1$. Describe an $O(n+m)$ time algorithm that computes a new maximum flow f', starting from the flow f for the modified flow graph. Justify the correctness of your algorithm.

Problem 8 (20 points) Vertex cut:

Let $G = (V, E)$ be a directed graph with distinguished vertices s and t. Describe an algorithm to compute a minimum sized set of vertices to remove to separate s and t. Your algorithm should identify the actual vertices to remove (and not just determine the minimum number of vertices that could be removed).

Problem 9 (20 points) Currency Conversion:

A group of traders are leaving India, and need to convert their Rupees into various international currencies. There are n traders and m currencies. Trader i has T_i Rupees to convert. The bank has B_j Rupees worth of currency j. Trader i is willing to trade as much C_{ij} of his Rupees for currency j. (For example, a trader with 1000 rupees might be willing to convert up to 700 of his Rupees for USD, up to 500 of his Rupees for Japaneses Yen, and up to 500 of his Rupees for Euros).

Assuming that all traders give their requests to the bank at the same time, describe an algorithm that the bank can use to satisfy the requests (if it can).