Final Exam
Computer Science 72700
Analysis of Algorithms
Dr. St. John
Graduate Center
City University of New York
18 December 2001

NAME (Printed) ____________________________
NAME (Signed) ____________________________
E-mail _________________________________

Exam Rules

• Show all your work. Your grade will be based on the work shown.
• The exam is closed book and closed notes.
• When taking the exam, you may have with you pens or pencils, and an 8 1/2” x 11” piece of paper filled with notes, programs, etc.
• You may not use a computer or calculator.
• Do not open this exam until instructed to do so.
• Please use a separate piece of paper for each problem!

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
1. (a) Write the **quicksort** algorithm for sorting a list of $n$ elements.
   (b) Analyse the worst-case behavior of this algorithm.
   (c) Write a **randomized** version of the quicksort algorithm.
   (d) Analyse the average-case behavior of this algorithm.

2. The **0-1 knapsack problem** is as follows: a thief robbing a store finds $n$ items; the $i$th item is worth $v_i$ dollars and weights $w_i$ pounds, where $v_i$ and $w_i$ are integers. He can carry at most $W$ pounds in knapsack and wants to take the most valuable load possible. What items should he take?

   In the **fractional knapsack problem**, the setup is the same, but the thief can take fractions of items, rather than having to make a binary (0-1) choice for each item.

   Both knapsack problems exhibit the optimal substructure property.

   (a) Show that the 0-1 knapsack problem does **not** have the greedy-choice property.
   (b) Show that the fractional knapsack problem has a greedy algorithm that is optimal (that is, it has the greedy-choice property).
   (c) What is the running time of the 0-1 knapsack problem? Justify your answer.
   (d) What is the running time of the fractional knapsack problem? Justify your answer.

3. The **diameter** of a tree $T = (V, E)$ is given by: $\max_{u,v \in V} \delta(u, v)$. That is, the diameter is the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm.

4. A **hamiltonian path** in a graph is a simple path that visits every vertex exactly once. Show that the language

   $$ \text{HAM-PATH} = \{ < G, u, v > | \exists \text{hamiltonian path from } u \text{ to } v \text{ in } G \} $$

   belongs to NP.

5. Consider a machine with $k$ fast memory locations and a large slow memory. The input is a sequence of page requests. If the page is fast memory, the request is satisfied. If it is not, a **page fault** occurs while some page is removed from main memory and the desired page is loaded into the empty spot. We measure the cost of an algorithm, $A$, on an input sequence $\sigma$, $C_A(\sigma)$ by the number of page faults required to service all the page requests.

   Let **FIFO** (First In First Out) be the algorithm that on a page fault, evicts the page that has been in memory the longest.

   Let **LRU** (Least Recently Used) be the algorithm that on a page fault, evicts the page in memory that was requested least recently.

   Let **LFU** (Least Frequently Used) be the algorithm that on a page fault, evicts the page in memory that was used least frequently.

   Fix the $k$, the number of fast memory registers. Prove or disprove:

   (a) FIFO is $k$-competitive.
   (b) LRU is $k$-competitive.
   (c) LFU is $k$-competitive.