Computer Science Department
Stanford University
Comprehensive Examination in Analysis of Algorithms
Autumn 1994

October 17, 1994

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in BLUE BOOKs. There are 5 problems in the exam. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. The number of POINTS for each problem indicates how elaborate an answer is expected. The total number of points is 60, and the exam takes 60 minutes. This "coincidence" can help you plan your time.

3. This exam is CLOSED BOOK.

4. Show your work, since PARTIAL CREDIT will be given for incomplete answers.

Comprehensive Exam: Analysis of Algorithms (60 points) **Autumn 1994**

1. **(5pts)** Determine the order of growth of the function $T(n)$, defined by the following recurrence:
   - $T(1) = 1$.
   - For $n \geq 2$, $T(n) = 3T(n/6) + n$.

2. **(5pts)** How many different irreflexive symmetric binary relations can be defined on a set of n elements ?

3. **(15pts)** Consider a weighted undirected graph with n nodes and m edges. Assume that you are given a minimum-weight spanning tree of this graph. How fast can you recompute the minimum spanning tree after the weight of one of the edges decreases by some given amount? Give a concise proof that your algorithm indeed produces a minimum-weight spanning tree.

4. **(15pts)** You are presented with a sequence of functions $f_1, f_2, \ldots$, where each function is defined on the interval from 1 to n. Function f_i is equal to 1 on the interval from 1 to some s_i and equal to 0 on the interval from $s_i + 1$ to n. The sequence is presented *one function at a time*. At any moment you might be given $x \in \{1 \ldots n\}$ and asked to evaluate at x the sum of all of the functions that were already revealed to you.
   Describe a data structure that stores the known functions and supports two operations: "evaluate $\sum_{i=1}^{k} f_i(x)$", where k is the number of known functions in the sequence, and "add a new function." One (inefficient) possibility is to maintain an array of size n. Initially, every element in the array is set to 0. Each time a new function (say, f_k) is revealed, we add 1 to all elements numbered 1 to s_k. In this case, adding a new function takes $O(n)$ operations and evaluating the sum takes $O(1)$ operations. Your goal is to design a data structure that is efficient both for computing the sum and for adding a new function to the set of known functions.

   Describe a data structure where each one of the above operations takes $O(\log n)$ time. Assume that you have $O(n)$ storage available.

5. **(20pts)** You have a computer that consistently trashes your disk each time you run more than k processes at the same time. You would like to find the value of k, while trashing your disk at most 2 times. Assume that you know that $k \leq n$.

   (a) **(5pts)** Describe a strategy that minimizes the number of "experiments" needed to find k, where an experiment consists of running some number of processes, running "fsck" afterwards to check whether the disk was trashed, and restoring the file system if it was.

   (b) **(5pts)** Generalize your strategy to any (constant) number of allowed crashes. Given a fixed number of crashes you are willing to sustain, what is the asymptotically maximum number of experiments used by your strategy ?

   (c) **(10pts)** Prove that, in the worst case, your strategy achieves asymptotically minimum possible number of experiments in the case where you are allowed only 2 crashes.