TESTS AND SOLUTIONS

Computer Science Department
Stanford University
Comprehensive Examination in Analysis of Algorithms

Autumn 1991

October 7, 1991

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in BLUE BOOKs. There are 4 problems in the exam; use a SEPARATE blue book for each problem. 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. You may not use any notes, calculators, computers, or outside help.
4. Show your work, since PARTIAL CREDIT will be given for incomplete answers.

"COMPS"
1991

Comprehensive Exam: Analysis of Algorithms (60 points) **Autumn 1991**
Please answer 3 out of 4 questions. If you attempt to answer all 4 questions, your grade will be the sum of the scores on the 3 best answers.

1. (20 points total) *Counting*
A group of n people comes to a party, each person carrying a hat and an umbrella. At the end of the party each person leaves with a hat and an umbrella, neither of which is his. Notice that in order to find out the number of possibilities, we can check all possible assignments of hats and umbrellas and count the number of appropriate assignments. The problem with this approach is that it takes exponential time. Can you come up with an approach that will allow us to compute the number of possibilities in polynomial time ? For example, an expression like $\sum_{i=1}^{n} i^3$ is an acceptable answer. In other words, your answer should be a formula computable in polynomial time.

2. (20 points total)
Given a sequence of numbers $a_1, a_2, \ldots, a_n$, a ***subsequence*** is a sequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$, where $i_j < i_{j+1}$ for all $1 \le j \le k-1$. You are given weights $w(a_i) \ge 0$ associated with each element of the given sequence. Describe an efficient algorithm to find an increasing subsequence of maximum weight, where the weight of the subsequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ is defined as $\sum_{j=1}^{k} w(a_{i_j})$. What is the running time of your algorithm? Explain.

3. (20 points total)
An ordered 2-3 tree T is used to implement a dictionary, with each element in the dictionary being assigned to a unique leaf in T. Initially, T is empty. Then the sequence of operations INSERT(a_1), INSERT(a_2), ..., INSERT(a_n) is performed, where each of the $n!$ possible orderings of the elements $a_1, a_2, \ldots, a_n$ is equally likely. Let h be the height of T following these n insertions (recall that a tree consisting of a single vertex has height 0).

   (a) (15 points)
   Give an exact formula for the maximum possible value of h (as a function of n).

   (b) (5 points)
   If $n = 30$, what is the expected value of h? [*Hint:* What are the minimum and maximum possible values of h?]

4. (20 points total) *Recurrence Relations*
Given any constant c, where $0 < c < 1$, and any positive real number N, the function $T(N, c)$ is defined as follows:

$$T(N,c) = \begin{cases} 2T(cN,c) + N^2 & \text{if } N > 1 \\ N^2 & \text{if } 0 < N \le 1 \end{cases}$$

   (a) (10 points)
   What is the asymptotic growth rate of $T(N, 1/2)$ (to within a constant factor)?

   (b) (10 points)
   What is the largest value of c such that $T(N,c) = O(N^2 \log N)$?

2