

Comprehensive Examination
Algorithms and Data Structures
Fall 2000

General directions: Answer 3 of the following 5 questions. Each problem is worth 10 points. Before giving more detailed descriptions of algorithms, describe in a brief (1-3 sentence) paragraph the main ideas and techniques. Complete pseudo-code is not necessary as long as you clearly specify how the algorithm works. You may use without proof any well-known algorithms and lower bounds for standard problems, as long as you state precisely and correctly the known result. Give at least an informal proof for all answers; for algorithms, this should include some convincing argument of correctness, and a time analysis. The time complexity of the algorithms you present will be given some weight in your score, as well as your correctness proofs and run-time analysis. For example, a correct $O(n^2)$ algorithm might be worth only 8 or 9 points if there is an $O(n \log n)$ algorithm. The exact weight of efficiency versus correctness is given after the specific problems.

I. Recurrence relations. Assume $a > 0, b > 1, c \geq 1$ are constants. Say that $T(n)$ is given by the recursion: $T(n) = c$ if $n < b/(b - 1)$ and $T(n) = n^a T(\lceil n/b \rceil)$ if $n \geq b/(b - 1)$. Will $T(n)$ be polynomial ($n^{\theta(1)}$), sub-polynomial ($n^{o(1)}$), or super-polynomial ($n^{\omega(1)}$)? Prove your answer.

II. Triangles in Graphs. Let G be an undirected graph with n nodes and m edges. A *triangle* in G is a clique of size 3, i.e., three nodes so that each pair is adjacent. a. (5 points; easy) Assume $m = o(n^2)$. Give an algorithm that is $o(n^3)$ time and outputs a list containing all triangles in G . b. (5 points, tricky) Assume $m = \theta(n^2)$. Describe an algorithm that *decides* whether G contains a triangle and is $o(n^3)$ time. (You do not need to give the exact time complexity of your algorithm, but need to state why it is $o(n^3)$.)

III. Decreasing Partitions. A decreasing partition of n is a decreasing sequence of positive integers whose sum is n . For example, 7;4;2;1 is a decreasing partition of 14. Give an algorithm that takes polynomial time in n to compute the number of decreasing partitions of n . (8 pts. correct poly-time algorithm, 2 points efficiency. $O(n^2)$ is pretty good.)

IV. Covering a spectrum. You want to create a scientific laboratory capable of monitoring any frequency in the electromagnetic spectrum between L and H . You have a list of possible monitoring technologies. $T_i, i = 1, \dots, n$, each with an interval $[l_i, h_i]$ of frequencies that it can be used to monitor. You want to pick as few as possible technologies that together cover the interval $[L, H]$. Give an efficient algorithm for finding such a set of technologies. (4 points for polytime algorithm. 3 points for correctness argument, 3 points for efficiency. $O(n^2)$ is relatively slow.)

V. Flows The hypercube is a graph with 2^n vertices given by n bit strings and where u is connected to v if you can obtain v from u by flipping exactly one bit. Say you want to solve the network flow problems in the hypercube, and you know that the capacities will be integers in the range from 1 to $\log n$. Which algorithm and data structures would you use? What would the running time be?