Computer Science Department
Stanford University
Comprehensive Examination in Analysis of Algorithms
Autumn 1993

October 29, 1993

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; 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.
4. Show your work, since PARTIAL CREDIT will be given for incomplete answers.
5. There is <u>only</u> one page of the exam.

Comprehensive Exam: Analysis of Algorithms (Solutions) Autumn 1993

1. (8 points)

   Determine (as tightly as you can) the order of growth of the function $T(n)$ defined by the following recurrence.

   - $T(1) = 1$.
   - For $n \geq 2$,

   $$T(n) = 2T(n/2) + \log n.$$

   Solution : Since we are interested in determining simply the order of growth, we may assume that $n = 2^k$ for some $k \geq 1$. We will show that $T(n)$ grows as $O(n)$. The recurrence may be rewritten as

   $$t_k = 2t_{k-1} + k$$

   where $t_k = T(2^k)$.

   We now unfold the recurrence to obtain

   $$t_k = 2^k + 2^k \sum_{i=1}^{k} \frac{i}{2^i}$$

   It is a trivial exercise to show that $\sum_{i=1}^{k} \frac{i}{2^i} = O(1)$. Alternatively, one may obtain a precise expression for $S(k) = \sum_{i=1}^{k} \frac{i}{2^i}$ by simply observing that $S(k) - \frac{1}{2}S(k) = \sum_{i=1}^{k} \frac{1}{2^i} - \frac{k}{2^{k+1}}$. It follows that $t_k = 3.2^k - k - 2$. Substituting $k = \log n$, we get $T(n) = 3n - \log n - 2$.

2. (5 points)

   Determine the number of binary sequences which contain n zeros and m ones such that each pair of ones is separated by at least k zeroes.

   Solution : Let S be the set of the binary sequences mentioned above and let S' be the set of binary sequences of length $n + m - k(m-1)$ which contain precisely m ones. For every sequence $\sigma \in S$, consider the sequence $f(\sigma)$ which is obtained by simply deleting k zeros between every pair of successive ones in σ. It is easily seen that f defines a bijection from S to S'. Thus the number of desired sequences is simply $\binom{n+m-k(m-1)}{m}$.

3. (10 points)

   Suppose you are given an array A containing n keys which are not necessarily distinct. In fact, you are guaranteed that there are only $O(\log n)$ distinct keys in the array. Show that you can sort A using $O(n \log \log n)$ time.

   Solution : We use variant of insertion sort where the keys are inserted into a balanced binary search tree data structure (like the AVL trees or the red-black trees). The slight modification is that each node in the tree has a counter associated with it. We insert

③

vertex of non-zero in-degree has a path leading into it from some vertex of zero in-degree. (Start at any non-zero in-degree vertex and keep moving back into any vertex with an edge coming into the current vertex. Since the graph is finite and acyclic, we will end up at a zero in-degree vertex.)

(b) (15 points) Describe an $O(|V| + |E|)$ time algorithm for finding a *minimum* start set in an *arbitrary* directed graph $G(V, E)$. Briefly justify the correctness of your algorithm.

Solution : Compute the strongly connected components of G, say $C_1, C_2, \ldots, C_k$ in $O(|V| + |E|)$ time. Construct the strongly connected component graph SCC which has a vertex corresponding to each strong component. There is an edge from a vertex i to j in SCC if and only if there exists $u \in C_i$ and $v \in C_j$ such that there is an edge from u to v in G. Clearly, the graph SCC is a directed acyclic graph. Now simply choose any vertex from each strongly connected component C_i such that the vertex i is a vertex of zero in-degree in SCC. It is easily seen that this set of vertices forms a start set, and the minimality of this set follows from the previous part of the question.