Comprehensive Exam: Algorithms and Concrete Mathematics Autumn 2004

This is a one hour closed-book exam and the point total for all questions is 60.

In questions that ask you to provide an algorithm, please explain the algorithm in words and diagrams, no need to write code or pseudo code. Also, for any algorithm, state and prove its running time. No credit will be given for exponential-time algorithms. Polynomial but slow algorithms will get some partial credit. Amount of credit will depend on how much slower they are compared to what is achievable using the knowledge in the reading list.

For full credit, the answers should be short and precise. Long and convoluted answers will not get full credit even if they are correct.

The following is a statement of the Stanford University Honor Code:

A. *The Honor Code is an undertaking of the students, individually and collectively:*
   (1) *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*
   (2) *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*

B. *The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*

C. *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

By writing my "magic number" below, I certify that I acknowledge and accept the Honor Code.

_____

(Number)

Prob	# 1	# 2	# 3	# 4	Total
Score					
Max	10	20	15	15	60

1. **[10 pts]** Prove a tight asymptotic bound on the behavior of $T(n) = 4T(\sqrt{n}) + \log n$, where $T(n) = \Theta(1)$ for $n < n_0$, where n_0 is a constants.

2. **[20 pts]** Your goal is to design a data structure for handling sets of positive integers (represented by arrays). It should support the following operations:

   - Given a set (represented by array of integers), add this set to the data structure. (e.g. ADD [1,7,2,13,55])
   - Given a set (represented by array of integers), detele this set from the data structure. For simplicity, assume that this set is legal, i.e. was added to the data structure at some point and was not yet deleted. (e.g. DELETE [7,550,78].)
   - Compute the size of the intersection of all of the sets in the data structure.

   Use n to denote the number of sets, k maximum number of elements in a set, and Q total number of distinct elements. For simplicity, assume that you know the values of Q, n, k in advance. Make sure that your memory usage is polynomial in the size of the input data.

   Make sure that dependence on Q is at most logarithmic. In other words, your data structure should be optimized for the case where Q and n are large, while k is relatively small.

   No need to write pseudo-code, just explain what techniques/elementary data structures you use and how you use them to implement each operation. You are allowed to use any data structures described in the reading list. In particular, it is ok to use arrays, lists, sorted lists, heaps, binary search trees, hashes, etc.

3. **[15 pts]** You are given a directed graph $G = (V, E)$ with n nodes and m edges, and two nodes s and t. Edges have non-negative weights. Given a path from s to t, redefine "length" of this path to be the sum of the lengths of the edges along the path *not counting the longest edge on this path.*
   Design an algorithm to compute shortest path from s to t using the modified definition of length. Faster algorithms will get more points. Make sure you state the asymptotic running time of your algorithm and provide proof that the algorithm is correct and that the running time is indeed as stated.

4. **[15 pts]** You are given an array of n integers from 0 to $n-1$, in sorted order starting from 0.

   - **[3 pts]** How much time will it take to convert this array into a heap ?
   - **[12 pts]** Once the array is converted into a heap, pick a random element and delete it from the heap, "fixing" the heap afterwards. What is the expected time of this operation ? Prove your answer. For simplicity, assume n is a power of 2.