# PhD Comprehensive Exam: Math and Algorithms

Fall 2000

The exam includes eight problems;
the length of the exam is three hours.

## Problem 1

Give proofs for the following asymptotic bounds:

**(a)** $\lg(n!) = \Theta(n \cdot \lg n)$

**(b)** $(\lg n)^2 = o(n)$

**Problem 2**

Give an asymptotic ($\Theta$-notation) solution for the following recurrence and show the derivation of your solution:

$T(n) = 2^n + \sum_{i=1}^{n-1} 2^{n-i-1} \cdot T(i)$

## Problem 3

Consider a binary heap, with the maximal element at the root, represented by an array $A[1..n]$. Suppose that you need to change the value of its $i$th element. Give an efficient algorithm that sets $A[i] = x$ and then updates the heap structure appropriately, that is, restores the heap property.

**Problem 4**

A sequence of numbers is called *increasing* if its elements are in sorted order; for example, $\langle 1, 2, 2, 3 \rangle$ is an increasing sequence. Write an efficient algorithm that determines the *length* of a longest increasing subsequence of an array $A[1..n]$; your algorithm does *not* have to find the longest subsequence itself. For example, if the input array is $\langle 1, 2, 1, 2, 3 \rangle$, then its longest increasing subsequence has 4 elements: $\langle 1, 2, 2, 3 \rangle$; thus, the algorithm must return 4.

**Problem 5**

Suppose that $G$ is an *undirected* graph, and you need to check whether $G$ has any cycle with an odd number of edges. Give an efficient algorithm that returns TRUE if $G$ has some odd-length cycle, and FALSE otherwise.

**Problem 6**

The classical Traveling Salesperson Problem requires to input a weighted graph and find a minimum-weight cycle that includes all vertices of the graph. We consider a modified version of this problem, which requires to find a minimum-weight *path* through all vertices; that is, the salesperson has to visit all vertices without returning to the initial vertex. The classical Traveling Salesperson Problem is known to be NP-complete. Use this fact to prove that the modified problem is also NP-complete.

## Problem 7

The Shortest-Cycle Problem requires to find a minimum-weight cycle in a given weighted graph. The cycle does *not* have to include all vertices; thus, this task is different from the Traveling Salesperson Problem. Determine whether the Shortest-Cycle Problem is NP-hard and justify your answer.

**Problem 8**

The *chromatic number* of an undirected graph is the minimal number of colors required for painting all vertices, in such a way that adjacent vertices have different colors. The problem of finding the chromatic number of a given graph is NP-complete.

Design a good approximation algorithm for this problem. It should find an "almost" minimal number of colors for a given graph, and output the corresponding coloring of vertices; its running time must be polynomial.

Give an example of a graph for which your algorithm does *not* find the correct chromatic number. If your algorithm is good, finding such an example should be difficult.

*Note: This problem is related to Problem 7(c) of the previous comprehensive exam; however, old Problem 7(c) asked for a simple coloring technique, which does not provide a good approximation.*