Algorithm (Ph.D. Candidacy)

1. (10%) Prove that $\lg(n!) = \Theta(n*\lg(n))$ and that $n! = o(n^n)$
2. (10%) What is the running time of heapsort on an array A of length *n* that is already sorted in increasing order? What about decreasing order?
3. (15%) Given an O(*n**lg(*n*))-time algorithm to find the longest monotonically increasing subsequence of a sequence of *n* numbers.
4. (15%) What is an optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?
   a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21
   Can you generalize your answer to find the optimal code when the frequencies are the first *n* Fibonacci numbers?
5. (10%) Given an algorithm that determines whether or not a given undirected graph G = (V, E) contains a cycle. Your algorithm should run in O(V) time, independent of |E|.
6. (10%) Suppose that the graph G = (V, E) is represented as an adjacency matrix. Given a simple implementation of Prim's algorithm for this case that runs in square of V time?
7. (10%) Show the Ford-Fulkerson algorithm for the maximal flow problem.
8. (10%) NP problems.
   (a) (3%) How can we prove that a problem is NP-hard?
   (b) (3%) How can we prove that a problem is NP-complete?
   (c) (4%) How can we prove that a problem is not NP-complete?
9. (10%) Randomized algorithm
   (a) (5%) Write a randomized algorithm to test whether a number is prime?

   (b) (5%) What is the probability that your answer is correct?