Questions for PhD Qualifying Exam in Algorithms

Fall 2013

DIRECTIONS: Please make sure to write very clear and easy to read answers.

PART I: REQUIRED:

A. Give a detailed and correct formal definition for the sets:

1. $O$

2. Omega

3. Theta

4. $P$

5. NP

6. NPC
B. Explain briefly the importance of efficiency in your PhD research. Does the research involve solving an optimization problem? Do you need your code to work in real time? Can the problem that you deal with have very large inputs? What are the bottlenecks that make efficiency hard to achieve? Do you need to design new algorithms?
PART II (EASIER QUESTIONS): PICK ANY AND ONLY THREE

1. You have to sort a file with 51,439,241,667 integers, with each integer being between 19,413 and 53,219. Your computer, by the way, has 1GB of RAM, but plenty of disk space. What strategy would you use to sort the numbers?

2. Integers larger than 2 can be made from the sum of three smaller strictly positive integers; for example, 3 = 1 + 1 + 1. 5 can be made from 3 + 1 + 1, and 2 + 2 + 1. Describe an algorithmic approach (and sketch some pseudocode) where you could find the number of different ways to form a value n from three smaller integers, assuming that you use a non-increasing order for the integers (in other words 3 + 1 + 1 is allowed, but 1 + 3 + 1 is not).

3. Assume you have a binary tree. Each vertex holds an integer value, the left child (if any) holds smaller values, and the right child (if any) holds larger values. Write a function that will return TRUE if all of the nodes rooted at a vertex V are between the values M and N (inclusive). The function should be as efficient as possible.
4. There are a number of different partitioning schemes for quicksort. Describe a few of the common variations, and what problems that might motivate you to choose one approach over another.

5. An algorithm can be described for a Universal Turing machine as a simple state diagram. Using the common “bubbles and arrows” formulation, assuming the paper tape holds an unsigned binary number, and that the tape reader is initially positioned at the lowest bit... Sketch a state diagram that will add 1 to the binary number.
PART III (HARDER QUESTIONS): PICK ANY AND ONLY THREE

1. You have a list of islands in the ocean, each with a specific XY location. You also have a boat, which requires fuel; each island can supply you with fuel, but you can travel a maximum distance of L on a tank. Describe your approach to each of these subproblems:
   a. Find the islands that you can safely travel between. This will be a sparse graph, and the method you use to find safe travel pairs should be efficient.
   b. Sketch or describe the algorithm you would use to go from some island A to some other island B, using the minimum fuel possible.
   c. If you are only interested in minimizing the number of stops (and not the fuel usage), describe the algorithmic approach. Is it more efficient, or less efficient, then when you were concerned with minimizing fuel?

2. You are in charge of a large project, with many workers, and many subtasks. For each task, there is a known amount of time for to complete it, and some tasks cannot be started until others have finished. Assume we have a graph G, where each vertex v_i represents a task which has a completion time t_i, and there are edges between the vertices to indicate ordering. Sketch pseudocode to determine the soonest that your project could be completed, assuming you have an abundance of workers.
3. You have bought a new car that has a rather unusual feature; each time a digit changes on the odometer, a bell rings. If, for example, the odometer reading goes from 00399 to 00400, the bell will ring three times.
   a. With a 6-digit odometer, what is the maximum number of times a bell can ring when you go from one reading to another?
   b. You have traveled 54,412 miles in this car. How many times has the bell rung?

4. The computer automated players for many two-player games such as chess, checkers, and tic-tac-toe use an artificial intelligence “minimax” approach. Describe an AI minimax approach for tic-tac-toe, and sketch the pseudocode for it.

5. Huffman coding is a key technique for compressing data. Assume you have a file with the following letters (and frequencies) – f:5, e:9, c:12, b:13, d:16, a:45. Give an optimal encoding for each of these letters to get the maximum compression possible.
6. When one needs to find all pairs of shortest paths (i.e., to have computed the shortest distance between any pair of vertices \(v_i\) and \(v_j\)), a popular approach is the Floyd-Warshall algorithm. The result of this algorithm is two matrices; one that holds the best distance between any pair of vertices, and a second vertex that holds the intermediate node (if any) between the pair. Sketch the algorithm in pseudocode, and explain how one would recover the actual path by using the matrix of intermediate vertices.

**PART IV (THEORY QUESTION):**

Pick any two NP-Complete problems you like, describe them, and show how one can be converted into another using a reduction. Include in your discussion the notion of a “certificate.”