國立台灣海洋大學資訊工程學系博士班

98學年度第一學期博士班資格考命題卷（筆試）

科目：演算法
命題教授：林清池老師
日期：2010/01/15

1. Please briefly describe the following standard terms and techniques which are commonly used in algorithm designs. (20%)

   (a) Dynamic Programming

   (b) Prune and Search

   (c) Divide and Conquer

   (d) Amortized Analysis

2. Let $f(n)$ and $g(n)$ be two asymptotically nonnegative functions. The asymptotic notations O, Ω, and Θ are defined as follows.

   - $f(n)$ is *asymptotically upper bounded by* $g(n)$, denoted by $f(n) = O(g(n))$, *iff* there exists positive constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.
   - $f(n)$ is *asymptotically lower bounded by* $g(n)$, denoted by $f(n) = \Omega(g(n))$, *iff* there exists positive constants c and n_0 such that $f(n) \geq c \cdot g(n)$ for all $n \geq n_0$.
   - $f(n)$ is *asymptotically equivalent to* $g(n)$, denoted by $f(n) = \Theta(g(n))$, *iff* there exists positive constants c_1, c_2 and n_0 such that $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ whenever $n \geq n_0$.

   Please answer the following questions according to the definitions given above. (25%)

   (a) Asymptotically upper bound the function $f(n) = n^2 + 6n + 1$ by the O notation. Justify your answer by demonstrating the constants.

   (b) Asymptotically lower bound the function $f(n) = n^2 + 6n + 1$ by the Ω notation. Justify your answer by demonstrating the constants.

   (c) Show that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

   (d) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/3 \rfloor) + n$.

   (e) Use the substitution method to verify your answer in (d).

3. Design an algorithm to find both the minimum and the maximum of a set of n elements with $3\lfloor n/2 \rfloor$ comparisons in the worse case (7%)

4. A *subsequence* is a sequence that can be derived from another sequence by deleting some elements. Given two sequences $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$, the ***longest common subsequence problem*** is to find a maximum-length common subsequence of X and Y. (12%)

   (a) Find an LCS of $\langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ and $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$.

   (b) Describe an algorithm that solves the longest common subsequence problem in $O(mn)$ time.

5. The *binary heap* data structure is an array object that can be viewed as a nearly complete binary tree. In a *max-heap*, we have $A[\textsc{Parent}(i)] \geq A[i]$ for every node i other than the root. (24%)

   (a) Show that an n-element heap has height $\lfloor \lg n \rfloor$.

   (b) Show that there are at most $\lceil n/2^{h+1} \rceil$ nodes of height h in any n-element heap.

   (c) Please give an algorithm that convert an n-element array A into a max-heap.

   (d) Briefly analyze the time complexity of your algorithm in (c).
   (*Hint:* You may use the results in (a) and (b).)

6. Let $G = (V, E)$ be an undirected graph and n be the number of vertices in G. (12%)

   (a) Please give an algorithm that solves the ***single-source shortest distance problem***. That is, an algorithm that computes the distances from a given vertex, say v_0, to all other vertices.

   (b) Briefly analyze the correctness and time complexity of your algorithm.