Ph.D. Qualifying Exam (Spring 2008)
Algorithms
Closed Book Examination

Question 1. (30%) (Radix Sort)

Using mathematical induction, prove carefully that radix sort works. Where does your proof need the assumption that the intermediate sort is stable?

Question 2. (30%) (Bucket sort)

We have n data which are uniformly distributed over $[0, 1)$. We divide the interval $[0, 1)$ into n equal-sized subintervals, or buckets. Next, the n data are distributed into the buckets, followed by running insertion sort on the data in each bucket. In the book, it is shown that the **expected** running time is linear.

Suppose we only have $n/2$ buckets (instead of n buckets). What is the increase in the expected running time compared to that of bucket sort with n buckets? State any assumptions that you need in arriving at the answer.

(A photocopy of the expected running time analysis for bucket sort taken from the 2nd Edition of the textbook 'Introduction to Algorithms' by Cormen, Leiserson, Rivest and Stein is provided.)

Question 3. (40%) (Dynamic Programming)

A subsequence is *palindromic* if it is the same whether read left to right or right to left. Note that a subsequence does not have to occupy consecutive locations in the original sequence. Let $a_1, a_2, \ldots, a_n$ be a sequence. Then a subsequence is $a_{\alpha_1}, a_{\alpha_2}, \ldots, a_{\alpha_k}$ where $1 \leq \alpha_1 < \alpha_2 < \alpha_3 < \ldots < \alpha_k \leq n$. For instance, the sequence

$$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

has many palindromic subsequences, including A, C, G, C, A and A, A, A, A. On the other hand, the subsequence A, C, T is not palindromic.

Devise an algorithm using dynamic programming that takes a sequence $a_1, a_2, \ldots, a_n$ where $a_i \in \{A, C, G, T\}$ for $1 \leq i \leq n$ and returns the longest palindromic subsequence. The running time should be $O(n^2)$.