Computer Science COMP-251B
Midterm, Feb 18, 2010, 14:35-15:55.
O P E N • B O O K S •/• O P E N • N O T E S

[25%] **1)**
Let

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 3T(\lceil n/4 \rceil) + T(\lceil n/8 \rceil) + O(n) & \text{if } n>1 \end{cases}$$

Prove by induction that T(n) is O(n).

[25%] **2) from Exercises 7.4-5**

The running time of quicksort can be improved in practice by taking advantage of the fast running time of insertion sort when its input is "nearly" sorted. When quicksort is called on a subarray with fewer than k elements, let it simply return without sorting the subarray. After the top-level call to quicksort returns, run insertion sort on the entire array to finish the sorting process. It may be argued that this sorting algorithm runs in $O(nk + n \lg(n/k))$ expected time. (but I am not asking you to argue it…)

How should k be picked, both in theory and in practice ?

[25%] **3) Exercises 5.2-5**

Let $A[1 .. n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an ***inversion*** of A. Suppose that each element of A is chosen randomly, independently, and uniformly from the range 1 through n. Use indicator random variables to compute the expected number of inversions.

[25%] **4) from Exercises 9.3-8**

Let $X[1 .. n]$ and $Y[1 .. n]$ be two arrays, each containing n numbers already in sorted order. We would like to find the median of all $2n$ elements in arrays X and Y.

a) Suppose the median of all the elements (of both X and Y) is in array X at position k. Express the conditions on X and Y that justify that $X[k]$ be the median of the $2n$ elements.

b) Give an $O(\lg n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y.
Hint: use the conditions from a), and use a binary search to find the element with the correct property.

Explain how your algorithm works and analyze the running time.