Analysis of Algorithms -Final

K. Subramani
LCSEE,
West Virginia University,
Morgantown, WV
{ksmani@csee.wvu.edu}

1 Instructions

1. The final exam is worth 30 points.
2. Exam Time: $8:00$ am $-$ $10:00$ am.
3. Attempt as many problems as you can. You will be given partial credit.
4. The solutions will be posted on the class URL on Friday, December 13.

2 Problems

1. Characterize the following recurrence: (5 points)

$$\begin{aligned} T(n) &= 1, \ \ if\ n = 1, \\ &= 9 \cdot T(\frac{n}{3}) + n^3 \cdot \log n, \ \ if\ n \geq 2 \end{aligned}$$

2. In the deterministic implementation of the Quicksort algorithm, we choose the last item as the pivot, at each level of the recursion. Assume that there exists an oracle, which when given an array of n elements, returns the middle element, i.e., the element of rank $\frac{n}{2}$, in $O(n)$ time. Now, instead of choosing the last element of the array as the pivot, suppose that we use the element of rank $\frac{n}{2}$ as the pivot in each level of the recursion. (You may assume that at each level, the number of elements is divisible by 2.) What is the running time of this modified version of Quicksort, on an array that is already sorted? (5 points)

3. Let **A** be an array of n *distinct* elements. If $i < j$ and $A[i] > A[j]$, for some indices i and j, then (i, j) is said to be an *inversion pair*.
   (a) List the inversion pairs of the array $\mathbf{A} = [2, 3, 8, 6, 1]$. (2 points)
   (b) Design a divide-and-conquer algorithm that outputs the *number of inversion pairs*, given an array having n elements. Analyze the time and space requirements of your algorithm. (8 points)
   Hint: Modify Merge-Sort.

4. Let $\mathbf{A} =< a_1, a_2, \ldots, a_n >$ be a sequence of integers, where each a_i can be positive, negative or zero. We define a con-sequence of **A** as a subsequence of **A**, with all the entries appearing consecutively in **A**. For instance, if $\mathbf{A} =< 7, -4, 3, 5, -2 >$, $< -4, 3, 5 >$ is a con-sequence. Note that $< -4, 5, -2 >$ is *not* a con-sequence, although it is a subsequence of **A**. The *worth* of a con-sequence is defined as the sum of its elements, e.g., the worth of $< -4, 3, 5 >$ is 4. Devise an algorithm that, given a sequence **A** of n elements, outputs the con-sequence of maximum worth. For instance, the con-sequence of **A** having maximum worth is $< 7, -4, 3, 5 >$ and it has worth 11. Analyze the time and space requirements of your algorithm. (10 points)
   Hint: Use Dynamic Programming.