DEPARTMENT OF MATHEMATICS, IIT - Kharagpur
Mid Semester Examination (Spring 2017)
MA 60002 Data Structure and Algorithm
No. of students: 130 Total Points: 30 DURATION: 2 Hours

---

Answer **ALL QUESTIONS**. All the notations are standard and no query or doubts will be entertained. If any data/statement is missing, identify it in your answer script. Marks are indicated at the end of each question.

1. Consider the recurrence $T(n) = T(n/2) + T(n/4) + n$. Use the substitution method to give a tight upper bound on the solution to the recurrence using $O$-notation. [2]

2. For each of the following algorithms, (i) give a recurrence that describes its worst-case running time and (ii) its worst-case running time using $\Theta$-notation: (a) Binary search, (b) Insertion Sort, (c) Merge Sort, (d) Randomized quicksort and (e) Strassen's algorithm. [5]

3. Consider the following sorting methods: Insertion Sort, Merge Sort, and Quick Sort. What is the running time using $O$-notation for each method

   (a) When all the the array values are equal?

   (b) When the values are in order?

   (c) When the values are in reverse order? [3]

   Explain your answers.

4. Consider the following outline of quicksort:

   **procedure** QuickSort(List);
   **begin**
     **if** (list has more than one item) **then**
       **begin**
       Choose a pivot element from the list;
       Partition list into two lists, L and R, using the chosen pivot.
       Sort L using QuickSort(L)
       Sort R using QuickSort(R)
       Return(QuickSort(L) followed by QuickSort(R))
       **end**
     **else** (Do nothing- list already sorted)
   **end**

   (a) What is the worst-case choice for a pivot?

   (b) What is the best-case choice for a pivot?

   (c) The median of a set of n numbers is a number $x$ such that at least $\lfloor \frac{n}{2} \rfloor$ numbers are at most $x$ and at least $\lfloor \frac{n}{2} \rfloor$ are at least $x$. In other words, if the numbers were to be sorted, the median would be in the middle of the list. Suppose that someone gives you a method FindMedian to find the median of n numbers in $O(n)$ time. How would you use FindMedian to improve the Quicksort method outlined above?

   (d) Write a recurrence relation for the worst-case running time for your new version of Quicksort.

   (e) What is the worst-case running time for the new version of quicksort? You should express your answer using $O$-notation. [5]

5. a) Is the sequence $\langle 20, 15, 18, 7, 9, 5, 12, 3, 6, 2\rangle$ is a max-heap? *Explain.* [2+2+3+2]

   b) Where in a max-heap can the smallest element reside, assuming all elements are distinct? Include both the location in the array and the location in the implicit tree structure.

   c) Suppose that instead of using `Build-Heap` to build a max-heap in place, the `Insert` operation is used $n$ times. Starting with an empty heap, for each element, use `Insert` to insert it into the heap. After each insertion, the heap still has the max-heap property, so after $n$ `Insert` operations, it is a max-heap on the $n$ elements.

      (i) Argue that this heap construction runs in $O(n \log n)$ time.

      (ii) Argue that in the worst case, this heap construction runs in $\Omega(n \log n)$ time.

   (d) Insertion sort can be expressed as a recursive procedure as follows. In order to sort $A[1\ ..\ n]$, we recursively sort $A[1\ ..\ n-1]$ and then insert $A[n]$ into the sorted array $A[1\ ..\ n-1]$. Write a recurrence for the running time of this recursive version of insertion sort.

6. **TRUE OR FALSE?** If the statement is correct, briefly state why. If the statement is wrong, explain why. [6]

   (a) By the master theorem, the solution to the recurrence $T(n) = 3T(n/3) + \log_2 n$ is $T(n) = \Theta(n \log_2 n)$.

   (c) There exists a comparison sort of 5 numbers that uses at most 6 comparisons in the worst case.

   (d) Heapsort can be used as auxiliary sorting routine in radix sort, because it operates in place.

   (f) Let $F_k$ denote the $k$-th Fibonacci number. Then, the $n^2$th Fibonacci number $F_{n^2}$ can be computed in $O(\log_2 n)$ time.

——-The End———