

Class Test 1  
Design and Analysis of Algorithms (January-April 2012)

1st February, 2012

**For each algorithm, write a correctness proof and analyze the time complexity.**

1. **(6 marks)** Arrange the following functions in nonincreasing order of asymptotic growth rate:  
 $n^{\log \log n}, \log n, 2^{\sqrt{2} \log n}, (\log n)^{\log n}$
2. **(6 marks)** Solve the following recurrences:
  - (a)  $T(n) = T(9n/10) + n$  by Recursion tree method
  - (b)  $T(n) = 4T(n/2) + n^2$  by Master method
  - (c) By substitution method, show that the solution to  $T(n) = 2T(n/2 + 17) + n$  is  $O(n \log n)$
3. **(5 marks)** Describe a  $\Theta(n \log n)$  time algorithm that, given a set  $S$  of  $n$  integers and another integer  $x$ , determines whether there exist two elements in  $S$  whose sum is exactly  $x$ .
4. **(5 marks)** Describe an algorithm to merge  $\frac{n}{k}$  sorted lists of  $k$  elements each.
5. **(8 marks)** Give an  $O(n^2)$  time algorithm to find a longest monotonically increasing subsequence of a sequence of  $n$  numbers.