Due Date: Saturday, 3 December 2005  
Total Mark: 55

**Problem 1:**  
[7 marks]

a) Design an algorithm for the **ternary search** which searches for a given key in a sorted array $A[1...n]$ in the same way as the binary search but in ternary search we split the given array in three parts instead of two parts and restricting the search to the appropriate piece.

b) Find the total number of key comparisons for the algorithm in part a) in the worst case and find its value when $n=1000000$.

**Problem 2:**  
[8 marks]

a) Write in (pseudocode) a divide and conquer algorithm for finding a position of the largest element in an array of positive integers.

b) Find the efficiency of the algorithm in part a) using master method.

c) Find the number of key comparisons in the worst case and best case.

**Problem 3:**  
[5 marks]

Find the total number of key comparison in the average case of the quick sort.

(2) If the partition can be happened in each position $i$ $(1 \leq i \leq n)$ with the same probability we will have $T_{avg}(n) = \frac{1}{n} \sum_{i=0}^{n-1} (T_{avg}(i) + T_{avg}(n-i-1)) + n$ and this is the same as $T_{avg}(n) = \frac{2}{n} \sum_{i=0}^{n-1} (T_{avg}(i)) + n$ which can be reduced to the recurrence relation $\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2}{n+1}$ and then use substitution (or telescoping method) with the approximation of the harmonic series to solve it.

**Problem 4:**  
[5 marks]

Design an algorithm to arrange elements in a given array of $n$ integers so that all its negative number precedes all its positive numbers, your algorithm should be in $O(n)$ time.

(Hint: can some idea we studied in quick sort help!!!)
Problem 5: [10 marks]

An array A[1... \( n-1 \)] contains all the integers from 1 to \( n \) with exactly one number \{1,2,..., \( n \) \} missing from A.

a) Write (in pseudocode) an algorithm to find the missing integer in \( O(n) \) time and \( O(n) \) additional space. (Hint: use an auxiliary array B[1... \( n \)] to record the elements of A).

b) Write (in pseudocode) an algorithm to find the missing integer in \( O(n) \) time and \( O(1) \) additional space, without ever changing the entries of A.

Problem 6: [5 marks]

Apply merge sort to sort the list: 7,9,4,13,1,9,4,8,6

Problem 7: [5 marks]

Apply quick sort to sort the list: 3, 10, 15,2,7,6, where the pivot is the last element in the list.

Problem 8: [10 marks]

A sorting algorithm is said to be parsimonious if it never compare the same pair of input value twice (Assuming that all input values are distinct). Which of the following sorting algorithm is parsimonious?

a- quick sort.
b- merge sort.
c- bubble sort.

for part a) justify your answer with either a counter example or a brief argument. (Did you realize why the quick sort is very efficient sorting algorithm in the average case?).

Problem 9: Bonus Problem (Optional) [7 marks]

Design decrease by half algorithm for computing \( \lfloor \log_2 n \rfloor \) and write its efficiency. (Hint: First prove that for any \( i > 1 \), there exist positive integer \( m \) such that 
\[ 2^m \leq i < 2^{m+1} \], then use it to show \( \lfloor \log_2 n \pm 1 \rfloor = \lfloor \log_2 n \rfloor \pm 1 \), \( n > 1 \), And then start writing your algorithm).