Chapter 7

1. Quick sort is an unstable sorting method. Give an example of an input list in which the order of records with equal keys is not preserved.

2. Write the status of the list 12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18. at the end of each iteration of the for loop of *insertionSort*.

3. What is the lower bound of the comparison based sorting algorithm? Justify your answer.

4. We have an input file as follows: F = (30, 9, 87, 6, 73, 17, 66, 20 48, 25)

   (a) Sort it by using merge sort in ascending order.

   (b) Sort it by quick sort in ascending order.

5. [*Bubble Sort*]In a bubble sort several left-to-right passes are made over the array of records to be sorted. In each pass, pairs of adjacent records are compared and exchanged if necessary. The sort terminates following a pass in which no records are exchanged.

   (a) Write a C function for bubble sort.

   (b) What is the worst-case complexity of your function?

   (c) How much time does your function take on a sorted array of records?

   (d) How much time does your function take on an array of records that are in the reverse of sorted order?

6. (a) Explain what ”a sorting method is stable” means.

   (b) Is Heap Sort a stable sorting method? If yes, please prove your answer. If no, please give an example to justify your answer.

7. Suppose you are given the following 10 numbers:55, 45, 25, 35, 85, 95, 65, 75, 105, 15.

   (a) Please sort these numbers in increasing order using merge sort. Please show necessary steps such that the use of merge sorting technique can be recognized.

   (b) Please sort these numbers in increasing order using quick sort. Please show necessary steps such that the use of quick sorting technique can be recognized.

8. Show how heap sort and quick sort for the following numbers work: 24, 76, 12, 41, 33, 5, 87, 80.

9. Please show the result of sorting 56, 6, 15, 100, 51, 38, 82 using radix sort with 7 buckets. The result of each pass must be listed.

10. Indicate both the stability and time complexity for each of the following sorting methods:

    (a) Insert sort

    (b) Quick sort

    (c) Merge sort

    (d) Heap sort