Introduction to Algorithms (CELEN086)

Problem Sheet 6

Topics: Recursion; Merge Sort; Bucket Sort

1. Trace algorithm merge() with two lists L1=[2,5,10,15] and L2=[3,6,7].

   What is the time complexity? Assume the original list before splitting has length n.

2. Show the process of sorting the list L=[9, 3, 15, 6, 8, 17, 10, 2] using merge sort.

   In general, if the list L has length n, how many "levels" of splitting are there?

3. Briefly explain why the Merge Sort has time complexity $O(n \log n)$.

   Note: in Seminar 6, we will design split() algorithm with time complexity $O(n)$.

4. Consider two lists:

   L1 = [24,26,22,29,66,11,20,27,21,23,25,28]

   L2 = [18,26,43,41,13,29,28,49,12,11,45,22]

   Which one is ideal for bucket sort? Select this list and show the process of bucket sorting.

5. Quick Sort is another divide-and-conquer scheme for sorting a list, with similar idea to Merge Sort. It has three components **partition**, **sort**(recursively) and **merge** as detailed in the following:

   - Partitioning: pick an element as the **pivot** (e.g., the first/last/middle element in the list). Then the list is partitioned in a way such that all elements less than the pivot are stored in the left sub-list while all elements greater than the pivot are stored in the right sub-list.
   - Sorting: for the left/right sub-lists after partition, if there are more than one elements, apply the Quick Sort scheme (partition-sort-merge) recursively for sorting each.
   - Merging: combine the three sub-lists after each partition, namely:

     [*left sub-list*] [*pivot*] [*right sub-list*].

   Sort the given lists in Q4 using Quick Sort. You should demonstrate it by showing necessary steps. Choose the front element as "pivot" for each partition.

   Note: time complexity of Quick Sort: $O(n \log n)$ *(best/average),* $O(n^2)$ *(worst).*