CS204: Algorithms

Mid Semester, Autumn 2018,

IIT Patna

Time: 2 Hrs

Full marks: 30

1. A subsequence of a given sequence is just the given sequence with zero or more elements left out. Formally, given a sequence X = <x_1, x_2, ..., x_m> another sequence Z =<z_1,z_2,z_3,...z_l> is a subsequence of X if there exists a strictly increasing sequence <i_1, i_2, ... ,i_k> of indices of X such that for all j = 1,2, ...k, we have x_{ij}=z_j . For example, Z =<A,B, C, D> is a subsequence of X =<A, B,C, B,D,A> with corresponding index sequence <2,3,5,7>. Given a sequence, we have to find the length of the longest palindromic subsequence in it. (2+2+4=8)

   a. Show that the problem exhibits optimal-substructure property
   b. Argue that problem also exhibits overlapping sub-problem property
   c. Develop an efficient algorithm to find out length of longest palindromic subsequence. Pseudo code and very brief explanation is expected.

2. Argue if the following statements are true or false. No marks will be awarded without proper justification. (6X2=12)

   a. Smallest element in a max-heap can be found in O(logn) time.
   b. Insertion sort is a stable sorting algorithm.
   c. Worst case of quick sort will take O(n^2) if pivot is chosen in such a way that every time it partitions the array in 1:9 ratio.
   d. Randomized partition algorithm partition the array in more balanced than 1:3 with 0.5 probability
   e. Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let T(n) be the number of comparisons required to sort n elements. Then T(n) is always <= T(n/5) + T(4n/5) +n
   f. We can sort using radix sort, if we apply quick sort for sorting each digit position.

3. Solve following recurrence relation (3X2=6)

   a. $T(n) = 2T(\sqrt{n}) + \lg n$
   b. $T(n) = 9T(n/3) + n$
   c. $T(n) = 7T(n/2) + n^2$

4. You are given a number say X, you have to represent X as sum of maximum number of composite numbers. For example if X=10 then it can be represented as 4+6. If X=12 then it can be represented as 4+4+4. If the number cannot be presented like this form then your algorithm should return 0. Like if X=7, then your algorithm should return 0. Develop an efficient algorithm for it. Pseudo code and very brief (4-5 lines) explanation is expected. (4)