JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

MID SEMESTER EXAMINATION-2015

B.Tech IV Semester

COURSE CODE: 10B11CI411   MAX. MARKS: 30

COURSE NAME: Fundamentals of Algorithms

COURSE CREDITS: 04   MAX. TIME: 2 HRS

Note: All questions are compulsory.

Section A   (Marks: 6)

1. Derive the complexity of build-heap algorithm?
2. What is the lower bound of any sorting algorithm?
3. If $f(n)$ increases with n, then for any positive integer k, $f(kn)$ is $O(f(n))$. Justify, whether it is true or false.
4. Show the red-black tree after successively inserting the keys 23, 19, 15, 12, 7 into an initially empty tree.
5. Where in a max-heap might the smallest element reside, assuming that all elements are distinct?
6. Prove that any comparison sort algorithm requires $\Omega(n \lg n)$ comparisons in the worst case.

Section B   (Marks: 9)

1. Given an array A of M+N elements. The first N elements in A are sorted and the last M elements in A are not sorted and information regarding to them is lacking.
   i. Evaluate the run-time complexity in term of M and N in the worst case, when using insertion sort on A?
   ii. For each of the following cases, which sort method (or methods combination) would you use and what would be the run-time complexity in the worst case?
   a) $M = O(1)$
   b) $M = O(\log N)$

2. Running time of an algorithm A and a competing algorithm B are described by the recurrences $T(n) = 3 T(n/2) + n$ and $S(n) = a S(n/4) + n$ respectively, assuming that $T(1) = S(1) = 1$ and $n = 4^k$ for some positive integer k, determine the value of a for which B is asymptotically faster than A.

3. Is the operation of deletion commutative in the sense that deleting x and then y from a binary search tree leaves the same tree as deleting y and then x? Argue why it is or give a counterexample.

1. Starting with the procedure MAX-HEAPIFY, write pseudo code for the procedure MIN-HEAPIFY *(A, i)*, which performs the corresponding manipulation on a min heap. How does the running time of MIN-HEAPIFY compare to that of MAXHEAPIFY?

2. What is the running time of QUICKSORT when all elements of array A have the same value? Prove that the average case run time of quick sort algorithm is O (n lg n).

3. Show that the longest simple path from a node x in a red-black tree to a descendant leaf has length at most twice that of the shortest simple path from node x to a descendant leaf. What is the largest possible number of internal nodes in a red-black tree with black height k? What is the smallest possible number?