Birla Institute of Technology & Science - Pilani, Hyderabad Campus
Second Semester 2017-2018
CS F211: Data Structures & Algorithms
Mid Semester Test

Type: Closed **Time: 90 mins** **Max Marks: 60** **Date: 06.03.2017**

All parts of the same question should be answered together.

1.a. Write an O(n)-time nonrecursive procedure that, given an n-node binary tree, prints out the key of each node. Use no more than constant extra space outside of the tree itself and do not modify the tree, even temporarily, during the procedure. [6 Marks]
1.b. Give an algorithm to find all nodes less than some value X in a binary heap. Analyze its complexity. [3 Marks]
1.c. Prove the average case complexity of bucket sort algorithm, as discussed in class, is O(n).
Note: Proof should be complete and all details of proof should be written. [6 Marks]

2.a. What is the running time of heapsort on an array A of length n that is already sorted in increasing order? What about decreasing order? [2 Marks]
2.b. Consider a generalization of the binary min-heap structure. Every node has d children. It is an almost complete,d-ary tree, and a node must be less than or equal to all its children. Design an array representation of the heap. Design a deleteMin and insertKey procedure here. [6 Marks]
2.c. Describe a recursive algorithm for enumerating all permutations of the numbers {1,2, ..,n}. What is the running time of your method? [6 Marks]

3.a. A common problem for compilers and text editors is determining whether the parentheses in a string are balanced and properly nested. For example, the string (((())())() contains properly nested pairs of parentheses, which the strings)()(and ()) do not. By making use of Stack as a data structure, can you give an algorithm that returns true if a string contains properly nested and balanced parentheses, and false if otherwise. If so devise an algorithm and find the complexity of the algorithm. Otherwise provide reasons for not being able to propose an algorithm. [5 Marks]
3.b. Suppose you are given an input set S of n numbers, and a black box that if given any sequence of real numbers and an integer k instantly and correctly answers whether there is a subset of input sequence whose sum is exactly k. Show how to use the black box O(n) times to find a subset of S that adds up to k. [6 Marks]

4.a. Assume you have an array A[1..n] of n elements. A majority element of A is any element occurring in more than n/2 positions (so if n = 6 or n = 7, any majority element will occur at least 4 positions). Assume that elements cannot be ordered or sorted, but can be compared for equality. (You might think of the elements as chips, and there is tester that can be used to determine whether or not two chips are identical.) Design an efficient divide and conquer algorithm of complexity O(n log n) to find a majority element in A (or determine that no majority element exists). [8 Marks]
4.b. Given any n, design an input of n elements such that the insertion sort takes $\Omega(n^2)$ operations. [2 Marks]

5.a. Suppose we are given a sequence S of n elements, each of which is an integer in the range $[0,n^2-1]$. Describe a simple method for sorting S in O(n) time. [8 Marks]
5.b. Is an array that is in sorted order always a min-heap? Also is an array that is in reverse sorted order (elements non-increasing) always a max-heap? [2 Marks]