Birla Institute of Technology & Science - Pilani, Hyderabad Campus
Second Semester 2015-2016
CS F211: Data Structures and Algorithms
Comprehensive Examination

Type: Closed **Time: 180 mins** **Max Marks: 100** **Date: 06.05.2016**

1.a. Answer the following questions with either True or False.
(i). Stack operations (push, pop, isEmpty) can be worst-case O(1) for a linked list implementation. [+1 0 -1]
(ii). Stack operations (push, pop, isEmpty) can be worst-case O(1) for an array list [+1 0 -1]
(iii). Enqueueing and dequeueing operations are all more expensive (worst case) for a priority queue backed by a heap, than for a regular queue backed by a linked list. [+1 0 -1]

1.b. Suppose that instead of doubling the size of an array-based stack when it becomes full, you increase the array size by the following sequence k, 2k, 3k, 4k, . . . for some positive constant k.
(i) If you have an empty stack that uses an array whose initial size is k, and you perform n pushes (assume that n > k), how many resize operations will be performed? [2 Marks]
(ii) What is the total cost complexity of executing n push operations? Explain your reasoning. [5 Marks]

1.c. An evil king has n bottles of wine, and a spy has just poisoned one of them. Unfortunately, they dont know which one it is. The poison is very deadly; just one drop diluted even a billion to one will still kill. Even so, it takes a full month for the poison to take effect. Design a scheme for determining exactly which one of the wine bottles was poisoned in just one months' time while expending only O (log n) royal tasters. State your scheme briefly, in English. [6 Marks]

2.a. 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. [4 Marks]

2.b. What is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number? [4 Marks]

2.c. We can sort a given set of n numbers by first building a binary search tree containing these numbers (using TREE-INSERT repeatedly to insert the numbers one by one) and then printing the numbers by an inorder tree walk. What are the worst-case and best-case running times for this sorting algorithm? [5 Marks]

2.d. Argue that since sorting n elements takes $\Omega(n \lg n)$ time in the worst case in the comparison model, any comparison-based algorithm for constructing a binary search tree from an arbitrary list of n elements takes $\Omega(n \lg n)$ time in the worst case. [5 Marks]

3.a. Suppose that each row of an n * n array A consists of 1's and 0's such that, in any row of A, all the 1's come before any 0's in that row. Assuming A is already in memory, describe a method running in O(n) time (not O(n2) time) for finding the row of A that contains the most 1's. [6 Marks]

3.b. An array A contains n integers taken from the interval [0, 4n], with repetitions allowed. Describe an efficient algorithm for determining an integer value k that occurs the most often in A. What is the running time of your algorithm? [4 Marks]

3.c. Suppose we have a hash table with 2n slots with collisions resolved by chaining, and suppose that n/8 keys are inserted into the table. Assume each key is equally likely to be hashed into each slot (simple uniform hashing). What is the expected number of keys for each slot? Justify your answer. [4 points]

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 in 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 a tester that can be used to determine whether or not two chips are identical.) Design an efficient divide and conquer algorithm to find a majority element in A (or determine that no majority element exists). Aim for an algorithm that does O(nlogn) equality comparisons between the elements. [8 Marks]

4.b. Suppose you are given an unsorted array A of all integers in the range 0 to n except for one integer, denoted the missing number. Assume $n = 2^k - 1$.

Design a O(n) Divide and Conquer algorithm to find the missing number. No marks will be given for non-Divide and Conquer algorithms. Argue (informally) that your algorithm is correct and analyze its running time. [8 Marks]

4.c. Consider another variation of the binary search algorithm so that it splits the input not only into two sets of almost equal sizes, but into two sets of sizes approximately one-third and two-thirds. Write down the recurrence for this search algorithm and find the asymptotic complexity of this algorithm. [4 Marks]

5.a. Consider a post office that sells stamps in three different denominations, 1p, 7p, and 10p. Design a dynamic programming algorithm that will find the minimum number of stamps necessary for a postage value of N pence. [6 Marks]

5.b. Consider the set of weighted intervals given below, where si is the start time, fi is the finish time, and vi is the value of the interval.

j	s_j	f_j	v_j	$p(j)$
1	0	6	2	0
2	2	10	4	0
3	9	15	6	1
4	7	18	7	1

Solve this instance of the weighted interval scheduling problem, i.e. find a set of (non-conflicting) intervals with maximum total weight. You should provide algorithm based on dynamic programming in a generic setting and then solve the problem for given instance. [8 Marks]

5.c. Solve the following instance of Fractional Knapsack problem with four items where the maximum allowed weight is $W_{max} = 10$. [2 Marks]

i	1	2	3	4
b_i	25	15	20	36
w_i	7	2	3	6

6. a. Suppose you're managing construction of billboards on the Rocky & Bull-winkle Memorial Highway, a heavily traveled stretch of road that runs west-east for M miles. The possible sites for billboards are given by numbers $x_1 < x_2 < \cdots < x_n$, each in the interval [0;M], specifying their position in miles measured from the western end of the road. If you place a billboard at position x_i, you receive a revenue of $r_i > 0$.

Regulations imposed by the Highway Department require that no two billboards be within five miles or less of each other. You'd like to place billboards at a subset of the sites so as to maximize your total revenue, subject to this restriction.

For example, suppose M = 20 and n = 5 with
$\{x_1, x_2, x_3, x_4, x_5\} = \{6, 7, 12, 13, 14\}$
and
$\{r_1, r_2, r_3, r_4, r_5\} = \{5, 6, 5, 3, 1\}$
Then the best solution is to place billboards at x_1 and x_3 to achieve a revenue of 10.
Describe a (dynamic programming) procedure to find a solution for this problem. What is the running time of your procedure (this should be polynomial in n)? [8 Marks]

Hint: As a first step towards the solution, define (similar to the Weighted Interval Scheduling Problem) e(j) to be the easternmost site that is more than 5 miles away from x_j. In other words, if you place a billboard at x_j, then $x_1; x_2; \colon\colon\colon; x_{e(j)}$ are also valid places to place billboards (subject to the same restriction about distances). Note that computing the e(j) values takes time O(n).

6.b. Suppose we want to make change for n cents, using the least number of coins of denominations 1; 10, and 25 cents. Consider the following greedy strategy: suppose the amount left to change is m; take the largest coin that is no more than m; subtract this coin's value from m, and repeat. Either give a counterexample, to prove that this algorithm can output a non-optimal solution, or prove that this algorithm always outputs an optimal solution. [4 Marks]

6.c. Devise an algorithm by using branch and bound technique to select items from the following table that maximizes benefit /value with the following constraints: [4 Marks]

1. The sum of weights of selected items should be less than or equal to 10
2. Items are indivisible, you either take an item or not.

Item	Weight	Value	Value / weight
1	4	$40	10
2	7	$42	6
3	5	$25	5
4	3	$12	4