Question Paper Analysis

1. Consider a hash table of size m = 7 and a corresponding hash function h(k) = k mod m. Compute the locations to which the keys 99, 59, 26, 50, and 58 are mapped using the following collision resolution techniques. (a) Chaining. (5) (b) Quadratic probing. (5)

Topic: Hashing Difficulty: 3

2. (a) Draw a Binary Search Tree by sequentially inserting the following elements: (4) 40,50, 22, 33, 30,80,15, 25, 60, 90, 75, 44 (b) For the BST obtained in

Topic: BST Difficulty: 6

3. (a), print the elements in Pre-order, In-order (3) and Post-order traversals. (c) Delete in sequence 44, 50, and 40 from BST obtained in

Topic: BST Difficulty: 9

4. (a). Show the (3) BST after each deletion.

Topic: BST Difficulty: 9

5. Define P and NP Complete class of problems. Give at-least one example for each (4) class.

Topic: P and NP Difficulty: 2

6. Show stepwise a tree that Huffman's greedy algorithm could produce for the (12) sentence given in Fig. 1. Write optimal codes thus generated for all the different characters present in the sentence (Fig. 1).

DATASTRUCTURESTRUCTURESDATA

Topic: String Difficulty: 6

7. Fig. 1 Note: Consider only those characters which are present in the given sentence. Use dynamic programming to fully parenthesize the product of four

matrices, (10) i.e. A [10 x 5], B [5 x 20], C [20 x 10], D [10 x 5], such that the number of scalar multiplications gets minimized. Show each and every step.

Topic: DP Difficulty: 7

8. Illustrate stepwise execution of the Heapsort algorithm on an array (10) A = [44, 30, 50, 22, 60, 55, 77, 55] to arrange its elements in descending order. 1/3

Topic: ARRAY Difficulty: 9

9. In n-Queens, some solutions are simply reflections of others. For example, two (10) solutions (Fig. 2) for 4-Queens problem are equivalent under reflection. (a) Modify NQueens Algorithm (Fig. 3) to get unique solutions only. (b) Execute the NQueens Algorithms devised in

Topic: BACKTRACKING Difficulty: 9

10. (a) for n = 1, 2, 3, and 4. Draw the associated state space search trees separately for each value of n. 1 1 2 2 3 3 4 4 Fig. 2 Fig. 3

Topic: BST Difficulty: 5

11. (a) Run Dijkstra's algorithm on the directed graph (Fig. 4), starting at vertex S. Show all the intermediate graphs in deriving the final shortest path tree. What is the order in which vertices get removed from the priority queue? (b) Write sequence in which nodes of the graph (Fig. 5) have been traversed using DFS and BFS, starting at vertex A. To make a unique solution, assume that whenever you faced with a dez.ision of which node to pick from a set of nodes, pick the node whose label occurs earliest in the alphabet. Fig. 4 Fig. 5

Topic: BST Difficulty: 9

12. (a) Let each node x in a binary search tree keeps an attribute x. successor that (6) points to x's inorder successor. Give a pseudocode for INSERT on a binary search tree T using this representation. (b) Given two integer arrays a[1.. m] and b[1.. n], find an integer that appears (6) in both arrays (or report that no

such integer exists). Assuming m n, give a crisp and concise algorithm in structured English satisfying following performance requirements: The amount of extra space (besides a[ ] and b[]) must be constant. It is fine to modify a[ ] and b[]. The worst case running time must be 0(n log m). Note: Algorithm will be evaluated on correctness, efficiency, and clarity. (5) (5) 2/3

Topic: ARRAY Difficulty: 5

13. (a) Fig. 6 presents an algorithm to find the minimum in a stack of integers (6) without affecting its contents. Determine if there is any bug in this code with respect to the following. Give suitable examples for each justification. i. Change in stack contents. ii. Incorrect answer. // S and temp are stacks. S contains some integers whose minimum is to be // determined. isEmpty(S) returns true if S is empty, else returns false. // min and t are integer variables. if (!isEmpty(S)) { min = pop(S); while (!isEmpty(S)) { t = pop(S); if (t < min) min = t; push(temp, t); while (lisEmpty(temp)) push(S,pop(temp); return min; } Fig. 6 (b) Consider a recursive static function f (), given below to answer the following (6) questions static int f(int n) if (n == 0) return 0; if (n == 1) return 0; if (n == 2) return 1; return f(n - 1) + f(n - 2) - f(n - 3); L What is the value of f (3)? H. What is the value off (4)? iii. What is the value of f (7)? iv. How many calls on f () are made to compute f (7), including the first one? v. What is the value of f (101)? ALL THE BEST 3/3

Topic: STACK Difficulty: 5