28/11/17

Indian Institute of Engineering Science and Technology, Shibpur

Five year Dual Degree (B.Tech-M.Tech) 3rd Semester Examination 2017

Data Structures and Algorithms CS-302

FULL MARKS: 70 TIME: 3 hrs.

Answer any five (5) questions. Students are advised to go through the questions carefully before answering. Clearly mention your assumptions, if any, while answering. Credits will be given to precise answer.

1. (a) Which of these following claims are correct?

(i) $(n+k)^m = \Theta(n^m)$, where k and m are constants.

(ii) $2^{n+1} = O(2^n)$

(iii) $2^{2n+1} = O(2^n)$

(b) Determine if $2n^3 + n^2 = \Theta(n^2)$.

(c) Ackerman Function is defined as follows.

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

Find $A(1,3)$. [4+6+4]

2. (a) Suppose a function is defined as follows:

$$S(n) = \begin{cases} 0 & \text{if } n = 0 \\ n + S(n-1) & \text{if } n > 0 \end{cases}$$

What would be the computing time of the above function?

(b) A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations *enQueue (add)* and *deQueue (delete)* can be performed in constant time?

(c) A single array $A[1 \ldots MAXSIZE]$ is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables *top_1* and *top_2* (*top_1* < *top_2*) point to the location of the topmost element in each of the stacks. Write down the procedures to *push()* and *pop()* element in the stacks mentioning clearly the condition for "stack full" and "stack empty". [4+5+5]

3. (a) Write down the complete analysis of running time of quick sort algorithm mentioning best, worst, and average cases.

(b) Is quicksort an unstable sorting method? Justify your answer with a suitable input instance.

(c) The insertion of an element in the insertion sort algorithm is carried out by (a) searching for the spot at which the insertion is to be made and (b) making the insertion. If as a result of the search it is decided that the insertion has to be made between R_i and R_{i+1}, then records $R_{i+1}, \ldots, R_n$ are moved one space to locations $R_{i+2}, \ldots, R_{n+1}$. Can the search in step (a) be sped up by incorporating binary search? Justify your answer. [6+4+4]

4. (a) Suppose there is an unsorted linked list in the memory. Suggest a suitable search procedure in which the frequently searched elements will be found with lesser number of comparisons. (You may modify the data structure if you wish.)

(b) Consider the following hash table of length 10 which uses open addressing with hash function, $h(k) = k\%10$, and linear probing.

0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

(c) Calculate the average number of comparison in successful and unsuccessful search if elements $42, 23, 34, 52, 46, 33$ are inserted in the order in a hash table of length 10 which uses the hash function $h(k) = k\%10$ and linear probing to resolve collision. [4+6+4]

5. (a) Suppose a binary tree T is in memory. Write a procedure to find the number of nodes in the tree.

   (b) Suppose the data items $A, B, \ldots, G$ are assigned the following weights:

$$(A, 13), (B, 2), (C, 13), (D, 23), (E, 29), (F, 5), (G, 9)$$

   Find the Extended binary tree with minimum weighted path length. What would be the Huffman coding for the 7 letters corresponding to the tree you have constructed?

   (c) *Postorder* traversal of a given *binary search tree*, T produces the following sequence of keys $10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29$. Write down corresponding *inorder* traversal of the tree T.

   (d) How many binary search tree of height 4 with 4 distinct values can be there? (Assume the root to be at height 1) [4+5+2+3]

6. (a) Prove that minimum number of nodes in an AVL tree of height h is $N_h = F_{h+2} - 1$, where $h \geq 0$ and F_{h+2} is the $(h+2)th$ Fibonacci number.

   (b) Write down an algorithm to merge the nodes of two AVL trees T_1 and T_2 together to obtain a new AVL tree. What is the computing time of your algorithm?

   (c) Assume that each node in an AVL tree T has the field *lsize*. For any node, a, $a \rightarrow lsize$ is the number of nodes in its left subtree + 1. Write an algorithm *avlFind(T,K)* to locate K-th smallest element in the subtree T. Show that this can be done in $O(\lg n)$ time if there are n nodes in T. [4+5+5]

7. (a) Construct a B-tree of order 3 with the the values $2, 30, 7, 15, 12, 10, 20, 4$ in oder. Show the steps clearly.

   (b) A *priority queue* is a collection of elements such that each element has been assigned a priority and such that the order in which elements are added and deleted comes from the following rules:

      (i) An element of higher priority is deleted before any element of lower priority.

      (ii) Two elements with the same priority are deleted according to the order in which they were added to the queue.

   Suggest a suitable data structure to represent the priority queue with procedures to *add()* a new element to and *delete()* an element from the *priority queue*. [6+8]