Paper Code: IC2216

Roll No. 4052

M.Sc (Informatics) II Sem.-2016
Paper-IT-22, Data Structure Design Of Algorithms

Time: 3hrs

Maximum Marks:75

(Write your Roll No. on the top immediately on receipt of this question paper)
(Answer Question 1 Compulsory and any 4 from the rest)

Q1)

(a) Take an initially empty hash table with five slots, with hash function h(x) = x mod 5, and with collisions resolved by ***chaining***. Draw a sketch of what happens when inserting the following sequence of keys into it: 35, 2, 18, 6, 3, 10, 8, 5.

(b) Repeat part (a) but with the following three changes: the hash table now has ten slots, the hash function is h(x) = x mod 10, and collisions are resolved by ***linear probing***.

(c) Write a function (Mention the function prototype clearly using 'C') which returns the median value from an integer array of n elements. What is the time complexity of your algorithm?

[3 * 5]

Q2)

(a) Suppose you have a list of numbers. You need to remove duplicates from that list, and also count the number of duplicates. Which data structure will be most efficient for doing this task?

After identifying the data structure, write pseudo codes to

1) Add an item in the data structure

2) Search an item

3) Delete an item from the Data Structure.

(b) Write a function (Mention the function prototype clearly using 'C') which accepts a string as argument, and returns 1 if all characters in the string are numeric, else it returns 0.

[9 + 6]

Q3)

Declare a suitable structure (in C language) that can represent a node of a Linked List (Assume you are storing an integer in each node).
Implement functions for the following tasks. (You need not write functions to get user data). Only write the proper function definitions. Also ***correctly specify the suitable argument(s) and return type for the functions.***

(a) Write a "***Count***" function that counts the number of times a given integer occurs in the list

(b) Write a ***GetNth()*** function that takes the linked list and an integer index and returns the data value stored in the node at that index position. GetNth() uses the 'C' numbering convention that the first node is index 0, the second is index 1, ... and so on. So for the list {42, 13,666} GetNth() with index 1 should return 13. The index should be in the range [0..length-1], where length is the number of nodes in the list.

(c) Write a function ***DeleteList()*** that takes the list, deallocates all of its memory and sets its head pointer to NULL (the empty list).

(d) Write a function ***InsertNth()*** which can insert a new node at any index within a list.

[4+4+4+3]

Q4)

(a) Consider you have a string to search "s" and you have a pattern "p". Write an algorithm that searches the pattern "p" in "s" and replaces it with another pattern "r". There can be multiple occurrences of pattern "p" in "s".
The final output of your algorithm will be the modified string say "t". In general while you are doing this, you need to figure out the algorithm which searches for a pattern in a string. How many ways you can do this?

(b) Write a function (in C language) that merges two arrays (consisting of integer elements) into one and returns a new array. Mention clearly what should be the prototype of the function that you have written.

[10 + 5]

Q5)

a) Find the solution of the following recurrence equations:

(i) $t_n = 4t_{n-1}, t_1 = 1 \; and \; n > 1$

(ii) $t_n = 4t_{n-1} - 3\,t_{n-2}\,, t_0 = 0\,, t_1 = 1 \; and \; n > 1$

(iii) $T(n) = 2T\left(\frac{n}{3}\right) + log_3 n, T(1) = 0$ *for* $n > 1$, *and* n *a power of* 3. [3*3]

b)

(*i*) Solve the following recurrence relation by substitution method.

$t_n = t_{n-1} + n, t_1 = 1$ *for* $n > 1$.

(*ii*) Suppose a complexity function $T(n)$ is eventually non decreasing and satisfies

$T(n) = aT\left(\frac{n}{b}\right) + cn^k, T(1) = d,$ *for* $n > 1$ *and* n *a power of* b.

where $b \geq 2$ and $k \geq 0$ are constant integers, and $a > 0, c > 0$, and $d \geq 0$ are constants. Then if $a = b^k$, show that $T(n) \in \Theta(n^k \lg n)$.

(*iii*) Suppose that $T(n)$ is eventually non decreasing and satisfies

$T(n) = 8T\left(\frac{n}{2}\right) + 5n^3, T(64) = 200,$ n a power of 2 and $n > 64$.

Find $T(n)$?

[3*2]

Q6)

(a) Using ***Binary Search Algorithm***, search for the integer 120 in the following list (array) of integers. Show the actions step by step.

12 34 37 45 57 82 99 120 134

(b) Use ***Quicksort*** to sort the following list. Show the actions step by step.

123 34 189 56 150 12 9 240

(c) Describe an algorithm to multiply two large integers. Use it to find the product of 1253 and 23103.

(d) Write the dynamic programming algorithm for the ***0-1 knapsack problem***.

[15]