

# College of Engineering, Pune

## End Semester Exam – November 2011

**S. Y. B. Tech. (Computer Engineering, Information Technology)**  
**(CT-203)- (Data Structures and Algorithms)**

**Day & Date-** Friday, 25<sup>th</sup> November 2011

**Maximum Marks:** 100

**Time:** - 8 am to 11 am

**Duration** – 3 hrs

**Instructions:**

1. All questions are compulsory. Numbers in the right-most column indicate marks.
2. Write indented and commented code.
3. This is an open book exam.
4. Make suitable assumptions if required, and state them in your answer.

<b>Q 1</b>	<b>A</b>	<p>Draw diagrams for following data structures (just draw the final diagrams, do not draw diagrams for intermediate stages)</p> <p>a) A doubly circular linked list of integers with following elements in it: 22 44 33</p> <p>b) A binary search tree of strings, with following sequence of insertions: what is your name</p> <p>c) A hash table with 5 buckets, linear probing used for resolving hash collision, with following hash function, and given data: 10 18 27  <code>int hash(int x) { return x &lt;&lt; 1 % 5; }</code></p> <p>d) An array based stack, with top initialized to the top end of array and stack growing towards index 0, after following order of push and pops: push(10), push(20), pop(), push(40), pop()</p>	<p><b>8</b> <b>(2</b> <b>*</b> <b>4)</b></p>
	<b>B</b>	<p>Given the following inorder and preorder traversals, reconstruct the binary tree of characters. Just draw the final diagram  <b>Inorder:</b> lehol <b>Preorder:</b> hello</p>	<b>4</b>
	<b>C</b>	<p>Write a function which checks if a given string is alphabetic or not. Do not use any library function.  <code>int isalpha(char *string);</code></p>	<b>4</b>
	<b>D</b>	<p>Draw diagram of the data structure created by the following code.</p> <pre>typedef struct elem {     char ch;     struct elem *ptr;     struct elem *chain; }elem; int main() {     elem p, q, r, s, t, *x;     p.ptr = r;      p.chain = q;     r.ptr = t;      q.chain = r;     t.ptr = q;      r.chain = s;     q.ptr = s;      s.chain = t;     s.ptr = p;      t.chain = p;     x = &amp;s;     return 0; }</pre>	<b>4</b>

Q2	A	<p>Draw Binary Search Tree after the given order of insertions and deletions. For deletion of a node with both subtrees, assume that the a node from the right subtree is chosen for replacement.</p> <p>Insert(30), insert(10), insert(50), insert(100), insert(20), insert(70), insert(4), insert(8), insert(65), delete(50), delete(8), delete(10)</p>	6
	B	<p>Show the status of the following sequence of numbers, after each iteration of bubble sort.</p> <p>56 89 32 14 100 98</p>	6
	C	<p>What is wrong with the code given below?</p> <pre>#define lim 128 int a[lim]; int i = 0; while(a[i] &lt; 20    i &lt; lim)     printf("%d ", a[i++]);</pre>	4
	D	<p>Derive an upper bound (O) for the following time complexity functions. Derive a bound as closer as possible.</p> <p>a) <math>F(n) = 10n^3 + 100n^2</math> b) <math>T(y) = y\sqrt{y} + y^2</math></p>	4
Q3	A	<p>Write a function to find median of a sequence of numbers stored in an array, using the partition function used for quicksort. Assume partition function is given to you, so no need to write code for partitioning.</p> <pre>int median(int a[], int len); // a is the array and len // it's length, the function returns the median value.</pre>	4
	B	<p>Write True or False with reasons.</p> <p>a) Quick sort is the best sorting algorithm available. b) In the worst case, a binary tree can degenerate into a linked list. c) EOF and NULL can be used interchangeably.</p>	6
	C	<p>For the following data, which of the given sorting techniques is best suited and why?</p> <p>data: 77 66 55 44 33 sorting techniques: insertion, selection, bubble</p>	6
	D	<p>Write a recursive function to destroy a singly linked NULL terminated list completely.</p> <pre>typedef struct node {     char ch;     struct node *next; }node, *list; void destroy(list *l);</pre>	4
Q4	A	<p>Write an implementation of a queue, using an array, such that enqueue always happens at index 0. Write enqueue, dequeue, init, empty, and full functions.</p>	6
	B	<p>Write a non-recursive function for a binary tree of integers, which finds out the average of all leaf node values. Assume availability of suitable data structures.</p>	8

		<pre>typedef struct node {     int val;     struct node *left, *right; } node, *tree; int leafaverage(tree t);</pre>	
	C	<p>Write a function which cuts a doubly circular linked list in the middle. After split the list remains as the first half, and the function returns the second part of the linked list as another doubly circular linked list.</p> <p><i>E.g. if the list was [10 20 30 40 50 60] , then after the function is called the list remains as [10 20 30] and the list [40 50 60] is returned by the function. For Odd No. of data elements, the bigger part remains in the original list.</i></p> <pre>typedef struct node {     int val;     struct node *prev, *next; }node; typedef struct dcLL {     node *head, *tail; }dcLL; dcLL cutmiddle(dcLL *l);</pre>	6
Q5	A	<p>Implement an integer linked list using an integer stack. (Note: This means, the only storage available to you is stack, no array is available. However you can use any number of stack variables and any number of elementary type variables in C. Stack is given to you, so don't write stack code). Implement only the following functions</p> <pre>void init(list *l); void traverse(list l); void insertatpos(list *l, int pos, int val);</pre>	6
	B	<p>Design data structure for a spread sheet. Just draw a diagram and write the type definitions in C.</p> <p><i>Spread sheet is a program which allows the users to store (assume) text data in cells of a sheet. Consider the following operations for the design: Entering data in a cell, editing data in a cell, sorting all or selected columns on a particular column, inserting and deleting a column or row. Popular examples of spread sheets are Libreoffice Calc or Microsoft Excel. You are expected to design data structure only for the storing the data and pay no attention to the graphical interface.</i></p>	8
	C	<p>You have been given an array of integers. The size is not known, so consider the size to be practically unlimited( which means do not bother about segmentation fault due to array index violations). The array is sorted in ascending order. Write code for an <b>efficient</b> function which finds a given number in the array and returns it's location.</p> <pre>int findno(int *a, int x); // x is the number to be found in array given by a.</pre>	6