



KIIT Deemed to be University
Online End Semester Examination(Autumn Semester-2021)

Subject Name & Code: Data Structure & Algorithms(CS-2001)
Applicable to Courses: B.Tech, Sem-3rd (Regular)

Full Marks=50

Time:2 Hours

SECTION-A(Answer All Questions. Each question carries 2 Marks)
Time:30 Minutes **(7×2=14 Marks)**

<u>Question No</u>	<u>Question Type (MC Q/SA T)</u>	<u>Question</u>	<u>CO Mapping</u>	<u>Answer Key (For MC Q Questions only)</u>
<u>Q.No:1</u>	<u>MC Q</u>	Let a two-dimensional array have a row range (45:90) and column range (50:205). The array is stored in row-major order. If the size of each element of this array is 2 bytes and base address of the array is 1000, then what will be the address of the element present at location (60,100). A) 5750 B) 5530 C) 5630 D) 5780	C0-1	D
	<u>MC Q</u>	Let a two-dimensional array have a row range (45:90) and column range (50:205). The array is stored in col-major order. If the size of each element of this array is 2 bytes and base address of the array is 1000, then what will be the address of the element present at location (60,100). A) 5750 B) 5530	C0-1	C

		C) 5630 D) 5780		
	<u>MC</u> <u>Q</u>	For one 2D matrix [15][20] each element size is 'W'. Let in column Major Order of storing, the address of [6][8] is 4440 and the base address of matrix is at [1][1] as 4000. Find the size of each element 'W'. A. 6 B. 4 C. 8 D. 10	C0-1	B
	<u>MC</u> <u>Q</u>	Let one n×n square matrix ARR is stored in the Column Major Order with element size as 4 bytes. If the base address is at ARR[1][1] as 1500 and the memory address of ARR[4][5] is 1608 then find out the size of matrix or value of n. a. 6×6 b. 12×12 c. 8×8 d. 16×16	C0-1	A
<u>Q.N</u> <u>o:2</u>	<u>MC</u> <u>Q</u>	What is the time complexity of fun() ? int fun(int n) { int count=0; for (int i= n; i> 0; i/=2) for (int j=0; j<i; j++) count+= 1; return count; } (A) $O(n^2)$ (B) $O(n \log n)$ (C) $O(n)$ (D) $O(n \log n \log n)$	C0-2	C
	<u>MC</u> <u>Q</u>	What is the time complexity of the below function? void fun(int n, int arr[]) { int i = 0, j = 0; for (; i< n; ++i) while (j < n && arr[i] < arr[j]) j++; } (A) $O(n)$ (B) $O(n^2)$ (C) $O(n \log n)$	C0-2	A

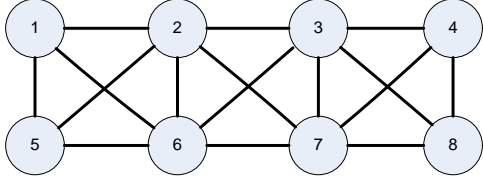
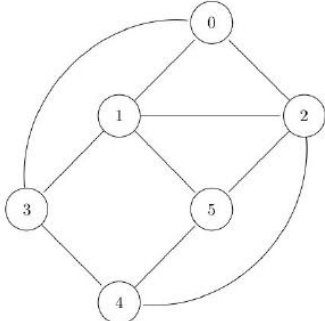
		(D) $O(n (\log n)^2)$		
	<u>MC</u> <u>Q</u>	<pre> int i, j, sum = 0; for (i = n / 2; i <= n; i++) { for (j = 2; j <= n; j = j * 2) { sum = sum + n / 2; } } for (i = 0; i < N; i++) { for (j = N; j < i; j--) { sum = sum + i + j; } } </pre> <p>The exact number of loop execution (in sequence of above code) and its time complexity</p> <ol style="list-style-type: none"> $(n/2)\log n + n^2$ and $O(n^2)$ $(n/2)\log n - 1 + n^2$ and $O(n^2)$ $(n/2 - 1)\log n + n^2$ and $O(n \log n)$ $(n/2)\log n + n$ and $O(n \log n)$ 	C0-2	D
	<u>MC</u> <u>Q</u>	<p>What is the time complexity of main()uj?</p> <pre> int fun (int n) { if (n==0) return 0; else return n+fun(n-1); } void main() { int i, n, sum; for (i=0; i<n; i+=2) sum+=fun(i); } </pre> <ol style="list-style-type: none"> $O(n^3)$ $O(n^2)$ $O(n \log n)$ $O(n^4)$ 	C0-2	A
<u>Q.N</u> <u>o:3</u>	<u>MC</u> <u>Q</u>	<p>Assume that the operators *, -, + are left to right associative and ^ is right to left associative. The order of precedence (from highest to lowest) is ^, *, +, -. What will be the postfix expression for the given infix expression $a+b*c-d^e^f$</p>	CO-4	A

		<p>(A) $abc*+def^{^^-}$ (B) $abc*+de^f^$ (C) $ab+c*d-e^f^$ (D) $-+a*bc^{^^}def$</p>		
	<u>MC</u> <u>Q</u>	<p>Assume that the STACK is initially empty. What will be the output of the following module.</p> <ol style="list-style-type: none"> 1. $XXX=2, YYY=5$ 2. PUSH (STACK, XXX) PUSH (STACK, 4) PUSH (STACK, $YYY+2$) PUSH (STACK, 9) PUSH (STACK, $XXX+YYY$) 3. WHILE TOP $\neq 0$ POP (STACK, ITEM) WRITE: ITEM END WHILE 4. Return <p>(A) 2, 4, 7, 9, 7 (B) 7, 9, 7, 4, 2 (C) 2, 4, 9, 11 (D) 7, 9, 7, 4</p>	CO-4	D
	<u>MC</u> <u>Q</u>	<p>Find the postfix expression of the given infix expression. Assume that \uparrow operator has the highest precedence and follows right to left associative.</p> <p>INFIX: $(a+b) \uparrow (p+q) \uparrow (r*s*t)$</p> <p>(A) $ab+pq+\uparrow rs*t*\uparrow$ (B) $ab+pq+\uparrow\uparrow rs*t*$ (C) $ab+pq+rs*t*\uparrow\uparrow$ (D) $ab+pq+rst**\uparrow\uparrow$</p>	CO-4	C
	<u>MC</u> <u>Q</u>	<p>Assume that \uparrow is the power operator and has the highest precedence. What is the output of the following postfix expression?</p> <p>$991*1\ 9\ \uparrow/+9-9+$</p> <p>(A) 19 (B) 10 (C) 18 (D) 81</p>	CO-4	C

<u>Q.N</u> <u>o:4</u>	<u>MC</u> <u>Q</u>	<p>Given the following sequence of letters and asterisks: EAS*Y*QUE***ST***IO*N***</p> <p>Consider the STACK data structure, supporting two operations PUSH and POP. Suppose that for the above sequence, each letter (such as E) corresponds to a PUSH of that letter onto the STACK and each asterisk (*) corresponds a POP operation on the STACK. What will be the sequence of values returned by the POP operations.</p> <p>(A) SYEUQTSAOINE (B) SYEUQTSAPONIE (C) SYEUQTSAPONI (D) SYEUQTSAPON</p>	CO-4	B
	<u>MC</u> <u>Q</u>	<p>Given the following sequence of letters and asterisks: EAS*Y*QUE***ST***IO*N***</p> <p>Consider the QUEUE data structure, supporting two operations INSERT and DELETE. Suppose that for the above sequence, each letter (such as E) corresponds to an INSERT of that letter into the QUEUE and each asterisk (*) corresponds a DELETE operation on the queue. What will be the sequence of values returned by the DELETE operations.</p> <p>(A) EASYQUESTOIN (B) EASYQUESTNIO (C) EASYQUESTNOI (D) EASYQUESTION</p>	CO-4	D
	<u>MC</u> <u>Q</u>	<p>Let Q be a QUEUE and S be a STACK. Assume that Q and S are initially empty. What is the last integer value printed by the given code ?</p> <pre> Enqueue (Q, 8); Enqueue (Q, 3); Push (S, 7); Push (S, 9); for (i=0; i<5; i++) { printf ("%d", Dequeue (Q)); printf ("%d", Pop (S)); Enqueue (Q, i); Push (S, i+5); } </pre> <p>(A) 4 (B) 9</p>	CO-4	C

		(C) 8 (D) 2		
	<u>MC</u> <u>Q</u>	<p>Let Q be a QUEUE and S be a STACK. Assume that Q and S are initially empty. What is the last integer value printed by the given code ?</p> <pre> Enqueue (Q, 8); Enqueue (Q, 3); Push (S, 7); Push (S, 9); for (i=0; i<5; i++) { printf ("%d", Pop (S)); printf ("%d", Dequeue (Q)); Enqueue (Q, i); Push (S, i+5); } </pre> <p>(A) 4 (B) 9 (C) 8 (D) 2</p>	CO-4	D
<u>Q.N</u> <u>o:5</u>	<u>MC</u> <u>Q</u>	<p>Hash table size=13, Hash function = $H(x) = x \bmod 13$, Collision resolution = quadratic probing = $h+i^2$ Keys=10, 100, 32, 45, 58, 126, 3, 29, 200, 400, 0, 21, 15. The key 15 will be stored at which location?</p> <p>A) 9 B) 10 C) 11 D) 12</p>	CO-5	D
	<u>MC</u> <u>Q</u>	<p>Suppose we are implementing quadratic probing with a hash function $H(x)=x \bmod 100$. If an element with key 9999 is inserted and the first three locations attempted are already occupied, then the next call that will be tried is:</p> <p>A) 99 B) 8 C) 4 D) 3</p>	CO-5	B
	<u>MC</u> <u>Q</u>	<p>Given the input {71, 23, 73, 99, 44, 79, 89} and a hash function $h(x)=x \bmod 10$ and quadratic probing, then in which location the last element will be placed?</p> <p>A) 2 B) 6 C) 7</p>	CO-5	D

		D) 8		
	<u>MC</u> <u>Q</u>	Given the input {37, 38, 72, 48, 98, 11, 56} and a hash function $h(x)=x \bmod 7$ with linear probing, then in which location key 11 will be placed? (The table size is 7 indexed from 0 to 6.) A) 3 B) 4 C) 5 D) 6	CO-5	C
<u>Q.N</u> <u>o:6</u>	<u>MC</u> <u>Q</u>	Which character will be placed in the root node of the expression tree for the following expression? $a/b-c/((d+e)-f)^g*h/p+k$ A) + B) - C) / D) ^	CO-4	A
	<u>MC</u> <u>Q</u>	Which character will be placed in the root node of the expression tree for the following expression? $a/b+c/((d+e)-f)^g*h/p-k$ A) + B) - C) / D) ^	CO-4	B
	<u>MC</u> <u>Q</u>	What is the post-order traversal of a binary tree whose in-order and pre-order traversals are as given below? INORDER: F, C, E, D, B, A, I, K, G, J, H PREORDER: A, B, C, F, D, E, G, I, K, H, J A) POSTORDER: F, E, D, C, B, K, I, J, H, G, A B) POSTORDER: F, C, E, D, B, K, J, I, H, G, A C) POSTORDER: F, C, E, D, B, K, G, I, H, J, A D) POSTORDER: A, C, E, D, B, K, J, I, H, G, F	CO-4	A
	<u>MC</u> <u>Q</u>	What is the pre-order traversal of a binary tree whose in-order and post-order traversals are as given below? INORDER: F, C, E, D, B, A, I, K, G, J, H POSTORDER: F, E, D, C, B, K, I, J, H, G, A A) PREORDER: A, B, C, F, D, E, G, I, K, J, H B) PREORDER: A, B, C, F, D, E, G, I, K, H, J C) PREORDER: A, B, C, F, D, E, G, K, I, H, J D) PREORDER: A, B, C, F, D, G, E, I, K, H, J	CO-4	B
<u>Q.N</u> <u>o:7</u>	<u>MC</u> <u>Q</u>	Let $G=(V, E)$ is an undirected graph where each edge has a unique cost. Consider the following statements and choose which one of the following is TRUE.	CO-4	C

		<p>I. For a given pair of vertices V_i and V_j, there always exist a unique shortest path between them.</p> <p>II. If the weights of the graph are multiplied by a positive constant, the shortest paths remain unchanged.</p> <p>A) I and II B) Only I C) Only II D) None</p>		
	MC Q	 <p>Identify valid BFS search sequences</p> <p>A) 1, 5, 2, 3, 4, 7, 6, 8 B) 1, 2, 5, 6, 7, 4, 3, 8 C) 1, 2, 5, 6, 7, 3, 4, 8 D) 1, 5, 2, 7, 6, 3, 4, 8</p>	CO-4	C
	MC Q	<p>Which of the following is an advantage of adjacency list representation over the adjacency matrix representation of a graph?</p> <p>A. In adjacency list representation, space is saved for sparse graphs.</p> <p>B. DFS and BSF can be done in $O(V + E)$ time for adjacency list representation. These operations take $O(V^2)$ time in adjacency matrix representation. Here V and E are the number of vertices and edges respectively.</p> <p>C. Adding a vertex in adjacency list representation is easier than adjacency matrix representation.</p> <p>D. All of the above</p>	CO-4	D
	MC Q	<p>Identify valid DFS search sequences. The search starts at vertex 0 and lexicographic ordering is assumed for the edges originating from each vertex.</p> 	CO-4	A

		A) 0 1 2 4 3 5 B) 0 1 2 5 4 3 C) 0 1 2 3 4 5 D) 0 1 3 4 2 5		
--	--	--	--	--

SECTION-B(Answer Any Three Questions. Each Question carries 12 Marks)

Time: 1 Hour and 30 Minutes

(3×12=36 Marks)

<u>Questi on No</u>	<u>Question</u>	<u>CO Mapping (Each question should be from the same CO(s))</u>
<u>Q.No: 8</u>	<p>Write the code to implement a queue using two stacks. [5]</p> <p>Apply the following sequence of operations stepwise in a normal queue as well as queue using two stacks. And analyze the performance in terms of their time complexity. [4]</p> <p>enqueue 1, 2, 3 dequeue dequeue enqueue 4, 5 dequeue</p> <p>Write the code to implement a priority queue where INSERT() will insert a letter with its priority value at the appropriate position and DELETE() will delete the letter with maximum priority inserted so far. Note: Consider the priority value of A to Z as 1 to 26. A higher number is having higher priority. [3]</p> <p>Write the code to implement a queue using a circular singly linked list. Show how you can perform enqueue and dequeue operations in O(1) time. [4]</p> <p>Write the code to implement a stack using two queues. [5]</p> <p>Apply the following sequence of operations in stack using two queues and stepwise show the output. [3]</p> <p>push 1, 2, 3 pop pop push 4, 5</p>	CO-4

	<p>pop</p> <p>Convert the given INFIX expression into POSTFIX expression using STACK. $5 * (((9 ^ 6 * 8 + 3) * (4 * 6)) + 7)$. [4]</p> <p>Evaluate the above POSTFIX expression using STACK. [4]</p> <p>Write the pseudo-code for evaluation of POSTFIX expression. [4]</p>	
<p>Q.No: 9</p>	<p>Given a single list, write the code to split the list into two sublists; one with prime numbers and the other with composite numbers. Then append the composite sub-list after the prime sub-list. [4]</p> <p>Construct an AVL tree by inserting the following elements in the order of their occurrence. Clearly mention which rotations you are applying in each step. 23, 34, 12, 11, 6, 2, 45, 4, 25, 24. [5]</p> <p>Write the INORDER, PREORDER, and POSTORDER traversal of the above AVL tree. [3]</p> <p>Given a single linked list, write the code to split it into two sublists; one for the 1st half, and other for the 2nd half. If the number of elements are odd, the extra element should go to the 1st sublist. After splitting, append the reversed 1st list after the 2nd list. Note: If the list is {2, 3, 5, 7, 11} then it should yield the two sublists {2, 3, 5} and {7, 11} and after append the output will be {7, 11, 5, 3, 2}. [4]</p> <p>Construct an AVL tree by inserting the following elements in the order of their occurrence. Clearly mention which rotations you are applying in each step. 50, 40, 35, 58, 48, 42, 60, 30. [5]</p> <p>Write the INORDER, PREORDER, and POSTORDER traversal of the above AVL tree. [3]</p> <p>Given two single linked lists, write the code to merge to make one list, taking nodes alternately between the two lists. So the output with {1, 2, 3} and {7, 13, 1} should yield {1, 7, 2, 13, 3, 1}. If either list runs out, all the nodes should be taken from the other list. [4]</p> <p>Construct an AVL tree by inserting the following elements in the order of their occurrence. Clearly mention which rotations you are applying in each step. 10, 20, 15, 25, 30, 16, 18, 19. [5]</p> <p>Write the INORDER, PREORDER, and POSTORDER traversal of the above AVL tree. [3]</p>	<p>CO3, CO-4, CO6</p>
<p>Q.No: 10</p>	<p>Suppose we know the preorder and postorder traversal sequences of a binary tree T. Can we uniquely determine the binary tree? Answer with a</p>	<p>CO-4, CO-6</p>

short justification. [2]
 Let the preorder traversal sequence of T be 100; 34; 16; 9; 8; 38; 11; 4; 81 and postorder traversal sequence be 34; 9; 11; 4; 38; 81; 8; 16; 100. If all the non-leaf nodes of T have two children, identify T. [4]

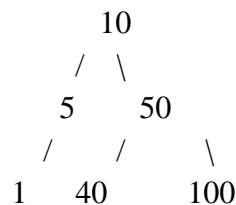
Illustrate the construction of 3-way B-Tree from the following sequence of elements. [3+3]

10, 60, 30, 20, 50, 40, 70, 80, 15, 90, 100, 85.

Delete 20, 60 and 50 from the constructed B-tree with proper clarification

Given a Binary Search Tree (BST) and a range low to high (inclusive). Write a non recursive code to count the number of nodes in the BST that lie in the given range. [4]

Input:

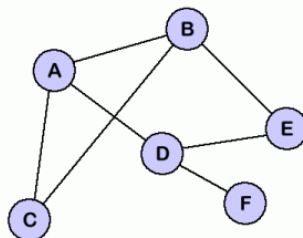


low = 5, high = 45

Output: 3 (5, 10, 40, are the node in the range.)

Let the preorder traversal sequence of a Tree be 48, 40, 33, 30, 35, 42, 58, 50, 60 and postorder traversal sequence be 30, 35, 33, 42, 40, 50, 60, 58, 48. If all the non-leaf nodes of the Tree have two children, identify the Tree and find the inorder traversal of it. [4]

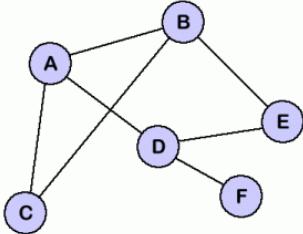
Write a procedure to find approachable nodes from a given source in a graph using stack as an intermediate data structure. Apply the above procedure to find the series of nodes which are approachable from node B [4]



Write anon-recursive code to find the average of minimum and maximum present in a binary search tree. [2]

Given a set of inserting values 10 through 16(7 continuous values). [2]

i. Write the sequence of values to be inserted in the BST such that the resulting tree will be a left-skewed tree. With diagram justify your

	<p>answer.</p> <p>ii. Write the sequence of values to be inserted in the BST such that the resulting tree will be a complete binary search tree. With diagram justify your answer.</p> <p>What is strictly binary tree. Write a code to convert one binary search tree to strictly binary tree by adding the absent child node (left/right) as follows. [4]</p> <p>i. If the left child of a node is absent then insert a new left child with half valued info of the node's info.</p> <p>ii. If the right child of a node is absent then insert a new right child with double valued info of the node's info.</p> <p>Write a procedure to find approachable nodes from a given source in a graph using queue as an intermediate data structure. Apply the above procedure to find the series of nodes which are approachable from node B [4]</p> 	
<p>Q.No: 11</p>	<p>Illustrate the steps to sort the numbers (in ascending order) 9, 2, 5, 6, 1, 4, 8 by using quicksort algorithm. [4]</p> <p>Show the step-by-step process to arrange the sequence of elements 5, 10, 50, 95, 15, 90, 30, 40 in ascending order using merge sort. [4]</p> <p>How many collisions occur if the hash addresses are generated using the modulo division method, where the table size is 64. 9893, 2341, 4312, 7893, 4531, 8731, 3184. [4]</p> <p>An array contains the following elements: 27, 30, 34, 45, 56, 59, 61. Show the step-by-step process to search 61 using binary search. Show the values of START, MID, and END in each step. [4]</p> <p>Illustrate the steps to sort the numbers (in ascending order) 47, 21, 23, 56, 12, 87, 19 by using quicksort algorithm. [4]</p> <p>How many collisions occur if the hash addresses are generated using the modulo division method, where the table size is 67. 9893, 2341, 4312, 7893, 4531, 8731, 3184. [4]</p> <p>Illustrate the steps to sort the numbers (in ascending order) 2, 5, 20, 34, 13, 19, 7 by using quicksort algorithm. [4]</p>	CO-5

	<p>An array contains the following elements: 12, 19, 23, 27, 30, 34, 45. Show the step-by-step process to search 45 using binary search. Show the values of START, MID, and END in each step. [4]</p> <p>Insert the following keys in an array of size 7 using the modulo division method. If collision occurs, then use linear probing to resolve the collisions. 94, 37, 29, 40, 84, 88, 102. [4]</p>	
--	---	--