Question Paper

Exam Date & Time: 22-Nov-2019 (02:00 PM - 05:00 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

INTERNATIONAL CENTRE FOR APPLIED SCIENCES
END SEMESTER THEORY EXAMINATIONS
NOVEMBER2019
III SEMESTER B.S. (ENGG.)
DATA STRUCTURES [ICS 231 - S2]

Marks: 100 **Duration: 180 mins.**

Answer 5 out of 8 questions.

1)	A)	Define the following terms: i) Time complexity ii) Space Complexity iii) Algorithm iv) Data structure	(8)
	B)	Write a recursive algorithm to find the number of binary digits in the binary representation of decimal number and compute the time complexity of the algorithm.	(6)
	C)	State and solve the recurrence relation of tower of Hanoi problem with backward substitution method. Initially, all the disks are on the first peg in order of size, the largest on the bottom and the smallest on top. The goal is to move all the disks to the third peg, using the second one as an auxiliary, if necessary.	(6)
2)	A)	Write a C++ program using template to construct a stack which includes push and pop operations.	(6)
	B)	Write an algorithm to evaluate postfix expression and evaluate the following expression. 7 6 8 + * 7 4 - 4 * +	(8)
	C)	Convert the following infix expression to postfix expression. A+((B-C*G)/K-D*F*Y) *T	(6)
3)	A)	Write an algorithm to implement circular queue which includes insert and delete functions.	(6)
	B)	Write a C++ program using template to implement a linked Queue with the following functions. i) INSERT ii) DELETE iii) DISPLAY	(8)
	C)	Write a C++ program using template to reverse the doubly linked list.	(6)
4)	A)	What is a binary tree. Draw a binary tree with 10 nodes labeled 0, 1, . . ., 9 in such a way that the inorder and postorder traversals of the tree yield the following lists: 9, 3,1, 0, 4, 2, 7, 6, 8, 5 (inorder) and 9, 1, 4, 0, 3, 6, 7, 5, 8, 2	(8)

(postorder). Justify your answer.

B) Construct an expression tree for the expression: C*(A+B) ^ (D*F*(G+H)). Convert it intro prefix and postfix notations. (8)

C) Write an algorithm to find the height of a binary tree. (4)

5) Write a C++ Program to construct a Binary Search Tree and display the tree in preorder. (8)

A)

B) Construct a Binary Search Tree for the list: 45 32 21 28 24 67 56 74 23 47. (6)

C) Construct an AVL tree for the list C, O, M, P, U, T, I, N, G. Use the alphabetical order of the letters and insert them successively starting with the empty tree. (6)

6) Devise an algorithm which divides its input elements according to their value. A partition is an arrangement of the array's elements so that all the elements to the left of some element A[s] are less than or equal to A[s], and all the elements to the right of A[s] are greater than or equal to it: (12)

A) A[0].....A[s-1] A[s] A[s+1].....A[n-1]
Compute the worst case time complexity of the method.

B) Write an algorithm which starts with A[1] and ends with A[n − 1], A[i] is inserted in its appropriate place among the first i elements of the array that have been already sorted. Sort the following list in alphabetical order using same method. (8)
E, X, A, M, P, L, E

7) For the input 30, 20, 56, 75, 31, 19, 34, 41, 89,78 and hash function h(K) = K mod 11 (8)

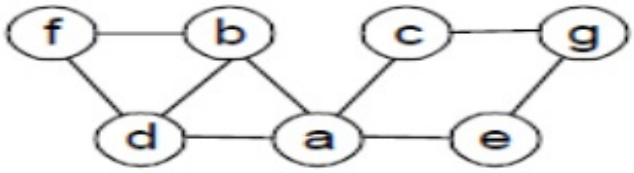
A)
i). Construct the closed hash table.
ii). Find the largest number of key comparisons in a successful search in this table.
iii). Find the average number of key comparisons in a successful search in this table.

B) Sort the following list in ascending order using heapsort method(Max-Heap). Show all the steps. (6)
2, 9, 7, 6, 5, 8

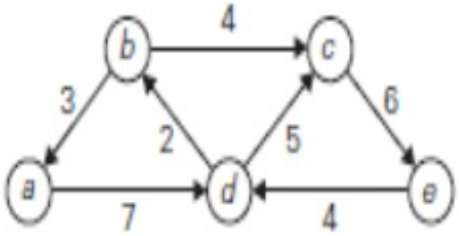
C) Compute the worst case time complexity of heapsort algorithm. (6)

8) Write the pseudocode for the Depth-First Search graph traversal. Consider the graph given below. Starting at vertex a and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search method and construct the corresponding depth-first search tree. (10)

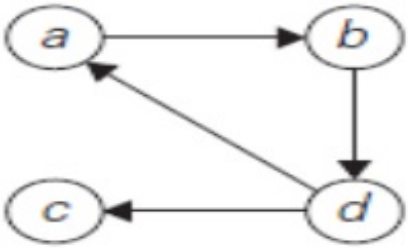
A)



B) Solve the following instances of the single-source shortest-paths problem with vertex **a** as the source: (4)



C) Apply Warshall's algorithm to find the transitive closure of the following digraph: (6)



-----End-----