CS3301 DATA STRUCTURES

IMPORTANT QUESTION

UNIT - I LISTS

2 - Mark

1. Define ADT. Give any two examples.
2. Define list. Mention any two operations that are performed on a list.
3. List out the areas in which data structures are applied extensively.
4. Define data structure with example.
5. What is circular linked list?
6. List out the advantage of circular linked list.
7. Distinguish between linear and nonlinear data structures.
8. Interpret the advantages and disadvantages of linked lists over arrays.
9. Differentiate arrays and linked lists.
10. Give an example for linked list application.
11. Examine a doubly linked list with neat diagram.
12. Illustrate the basic operations carried out in a linked list.
13. Show the ways in which list ADT can be implemented.
14. Compare calloc() and realloc() function and mention its application in linked list.
15. Analyze and write a routine to find position of given element in singly linked list.
16. Analyze and write the linked list representation of a polynomial:
    $p(x) = 4x^3 + 6x^2 + 7x+9$
17. Should arrays or linked lists be used for the following types of applications? Support your justification.
    i. Many search operations in sorted list.
    ii. Many search operations in unsorted list.
18. Compare between linear linked list and circular linked list.
19. Design a routine to delete an element in a linked list.
20. Develop an algorithm for insertion operation in a singly linked list.

13 - Mark

1. Explain Abstract Data Types (ADTs).
2. Describe Array-based implementation.
3. Describe Linked list implementation.
4. Write about Singly linked lists.
5. Explain Circularly linked lists.
6. Explain Applications of lists.
7. Write detailed notes for Polynomial ADT.

UNIT - II STACKS AND QUEUES

2 - Mark

1. Define stack and specify the operations
2. List any four applications of stack?
3. Given the prefix for an expression. Write its postfix:
   ++A*BCD and +*AB*CD4
4. Given a infix expression convert it into postfix expression using
   stack a+b*(c^d-e)^(f+g*h)-i
5. Discuss the postfix and prefix forms of the expression?
   A+B*(C-D) / (P-R)
6. Illustrate the purpose of top and pop?
7. What causes underflow of stack? How it could be avoided?
8. How to implement stack using linked list.
9. Summarize the rules followed during the infix to postfix conversions.
10. Generalize a routine to return top element in a stack using linked list11
11. Define double ended queue.
12. List the applications of a queue.
13. What are priority queues? What are the ways to implement priority queue?
14. What is circular queue?
15. Circular queue is better than standard linear queue. Why?
16. Classify the different types of queues.

17. Show a routine to perform enqueue operation in a queue.
18. Differentiate between double ended queue and circular queue.
19. For railway reservation the queue data structure is preferred -Justify
20. Develop an algorithm for deleting an element in a double ended queue.

13 - Mark

1. Describe about stack ADT using array in detail.
   (i) Give an algorithm for push and pop operations on stack using a linked list with an example.
   (ii) Describe the function to examine whether the stack is full () or empty ().
2. Explain Balancing Symbols.
3. Describe Evaluating arithmetic expressions.
4. Write about Infix to Postfix conversion
5. Explain Applications of Queues.

UNIT - III TREES

2 – Mark

1. If the depth of binary tree is k, the maximum number of nodes in the binary free is 2"-1. Prove.
2. Recommend the result of inserting 3,1,4,6.9.2.5,7 into an initially empty binary search tree.
3. Deline a binary tree. Give an example.
4. Create an expression tree for the expression. ((a + ((b/c)*d))- e)
5. Differentiate AVI. tree and Binary search tree.
6. Give the various types of rotations in AVL tree during the insertion of a node.
7. What are threaded binary trees? Give its advantages
8. Define balance factor of AVI. Tree.
9. Simulate the result of inserting 2, 1. 4, 5, 9, 3, 6, 7 into an initially

empty AVL Tree.

10. Define an expression tree. Give an example for it.
11. Summarize tree traversal and mention the type of traversals.
12. Differentiate B tree and B+ tree
13. Point out the properties of B+ tree.
14. Illustrate the benefits of B+ tree.
15. List out the various operations that can be performed on R-trees
16. Identify the structural properties of B-Trees.
17. Illustrate the steps in the construction of a heap of records with the
18. following key values: 12. 33. 67, 8. 7, 80, 5. 23.
19. Analyze the properties of binary heap.
20. Define a heap and show how it can be used to represent a priority queue.

13 - Mark

1. Write an algorithm for preorder, inorder and postorder traversal of a binary tree.
2. Explain the following operations on a binary search tree with suitable algorithm

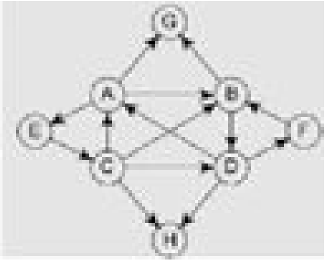
   i) Find a node

   ii) Find minimum and maximum elements of BST.

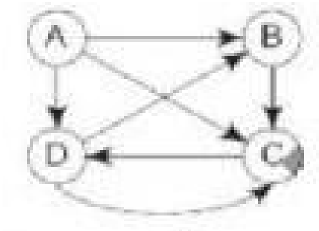
UNIT - IV MULTIWAY SEARCH AND GRAPHS

2 – Mark

1. Define graph.
2. Consider the graph given below. Create the adjacency matrix



3. Find out the in-degree and out-degree of the given graph



4. Create an undirected graph and its adjacency matrix for the following specification of a graph G.

   V(G)=1,2,3.4

   G= {(1,2), (1,3), (3,3), (3,4), (4,1)}
5. Differentiate BIS and DIS.
6. What is meant by bi-connected graph?
7. Give the purpose of Dijkstra's algorithm.
8. Differentiate cyclic and acyclic graph.

13 - Mark

1. Compare Breadth-first traversal & Depth-first traversal
2. Describe Bi-connectivity.
3. State Euler circuits.
4. Explain Dijkstra's algorithm.
5. Write about Minimum Spanning Tree.
6. Explain Prim's algorithm
7. State Kruskal's algorithm

UNIT - V SEARCHING, SORTING AND HASHING TECHNIQUES

2 - Mark

1. What is hashing?
2. Define extendible hashing and give its significance.
3. What is meant by internal and external sorting? Give any two examples for each type.
4. List the different types of searching
5. Define rehashing.
6. Identify the advantage of shell sort over insertion sort.
7. How many passes does the of insertion sort algorithm do to sort a list of 5 elements? What happens in its i" pass?
8. Give the types of collision resolution.
9. Interpret the fastest searching algorithm and give reason.
10. Distinguish between linear and binary search technique.
11. Classify the different sorting methods.
12. Apply insertion sort and sort the following elements 3,1,4,1,5.9.2,6.
13. Which hashing technique is best and illustrate with an example?
14. Analyze why do we need a hash table as a data structure as compared to any other data structure?
15. Point out the advantages of using open addressing.
16. Compare the advantage and disadvantage of separate chaining and linear probing?
17. Select the best sorting method out of the following - insertion sort quick sort and merge sort and give justification.
18. Summarize the open addressing hashing method with an example.
19. Develop an algorithm for a shell sort.
20. Prepare a simple C Program for a linear search.

13 - Mark

1. Describe about selection sort with suitable example.
2. Examine the algorithm for Insertion sort and sort the following array:39.9.45.6318,81, 108.54,72,36
3. List the different types of hashing techniques. Explain them in detail with an example.
4. Show the result of inserting the keys 2, 3, 5, 7, 11, 13, 15, 6, 4 into an initially empty extendible hashing data structure with M = 3.
5. Write a C program to search a number with the given set of numbers using binary search.
6. Interpret an algorithm to sort a set of 'N' numbers using bubble sort and demonstrate the sorting steps for the following set of numbers: 30. 52, 29, 87, 63, 27, 19, 54.
7. Discuss the various open addressing techniques in hashing with an example.
8. (i) Sort the given integers and show the intermediate results using shellsort:39.9.81.45,90,27,72,18.

   (ii) Write an algorithm to sort an integer array using shell sort