CS3301 DATA STRUCTURES
QUESTION BANK

UNIT I -LISTS

Abstract Data Types (ADTs) – List ADT – Array-based implementation – Linked list implementation – Ssingly linked lists- Circularly linked lists- Doubly-linked lists – Applications of lists –Polynomial ADT– Radix Sort – Multilists

PART-A

1. Define ADT. Give any two examples.
2. What is the advantage of an ADT?
3. Define data structure.
4. How data structures are classified?
5. What are the ways of implementing linked list?
6. Distinguish between linear and non linear data structures.
7. List the various operations that can be performed on data structure.
8. Compare calloc() and realloc() function and mention its application in linked list.
9. Describe the differences between singly and doubly linked lists.
10. List out the areas in which data structures are applied extensively.
11. Define non linear data structure.
12. What are the advantages of modularity?
13. What is dynamic allocation?
14. What is a linked list?
15. What is the need for the header?
16. List three examples that uses linked list?
17. Compare singly linked list with circular linked list.
18. How the singly linked lists can be represented?
19. How the doubly linked list can be represented?
20. What are the types of linked list?
21. What are the pitfall encountered in singly linked list?
22. State the advantages of circular lists over doubly linked list.
23. What are the advantages of doubly linked list over singly linked list?
24. Define doubly linked list.
25. Design a routine to delete an element in a linked list.
26. Compare calloc() and realloc() function and mention its application in linked list.
27. Analyze and write the linked list representation of a polynomial: p(x)=4x3 +6x2 +7x+9.
28. Write down the steps to modify a node in linked lists
29. List out the advantage of circular linked list.
30. Binary search cannot be performed on a linked list. Examine.
31. Discuss the advantages and disadvantages of linked lists and arrays.
32. Give an example for linked list application.
33. Specify the use of Header node in a linked list.
34. Illustrate the use of linked list with an example.
35. Show the ways in which list ADT can be implemented.

36. Differentiate arrays and linked lists.
37. State the properties of LIST abstract data type with suitable example.
38. Analyze and write a find routine in array implementation.
39. Why is the linked list used for polynomial arithmetic?
40. What are the postfix and prefix forms of the expression: A + B* (C – D) / (P – R)
41. Analyze and write the array representation of a polynomial: p(x) = 4x3+6x2+7x+9
42. Should arrays or linked lists be used for the following types of applications? Support your justification. 1. Many search operations in sorted list. 2. Many search operations in Unsorted list.
43. Develop an algorithm for insertion operation in a singly linked list.
44. Define Radix Sort
45. Define Multilists

PART – B

1. Explain the insertion operation in linked list. How nodes are inserted after a specified node.
2. Write an algorithm to insert a node at the beginning of list?
3. Describe the following:
   i. Applications of lists. (5)
   ii. Polynomial manipulation. (8)
4. What is a linked list? (2)
   ii. Describe the suitable routine segments for any four operations. (11)
5. List an algorithm to perform the following operations in a doubly linked list.
6. i. Insert a node at the end of the list.(6).
   ii. Delete the last node in the list. (7)
7. i. Discuss the insertion and deletion procedures for cursor based linked lists.(7)
   ii.Give an algorithm for the deletion and reverse operations on doubly linked list. (6)
8. i. Give the algorithm to perform insertion on a doubly linked list.(7)
   ii. Give the algorithm to perform deletion on a doubly linked list.(6)
9. Write an algorithm to demonstrate a polynomial using a linked list for
   i. Addition and Subtraction. (7)
   ii.Multiplication operations. (6)
10. Write the program for array implementation of lists
11. Analyze and write algorithm for Circular Linked list for the following operations using structure pointer.
    i.Create & Insert . (6)
    ii. Delete & Display.(7)
    iii. Merge operation.
12. Explain the application of linked list in detail.
    i. Radix sort. (7)
    ii. Multi list. (6)
13. Consider an array A[1: n] Given a position, write an algorithm to insert an element in the Array. If the position is empty, the element is inserted easily. If the position is already occupied the element should be inserted with the minimum number of shifts. (Note: The elements can shift to the left or to the right to make the minimum number of moves). (13)
14. Develop a program to add the values of the nodes of a linked list and then calculate the mean. (13)

15. Describe the various operations of the list ADT with examples. (13)
16. Illustrate how polynomial manipulations are performed using lists? Explain any two operations with example. (13)
17. Explain the steps involved in the following insertion operations in a singly linked list.
    i.Insert the node in the start and End. (7)
    ii.Insert the node in the middle of the List (6)
18. Dicuss an algorithm for linked list implementation of list. (13)
19. Write an algorithm to implement radix sort with suitable example. (13)
20. Explain in detail about Multilists with example. (13)

PART – C

1. Create an algorithm to add two polynomials using linked list.(15)
2. Explain an algorithm to split a linked list into two sub lists containing odd and even ordered elements in them respectively.(15)
3. Explain an algorithm to merge two sorted linked lists into a single sorted list.(15)
4. Design algorithm for various operations performed on circular linked list. Extend the algorithm defined in the previous question for the doubly linked circular list. (15)
5. Recommend an algorithm to add two polynomials 5x2 +3x+15,2x2 +6x+3 when the polynomials are represented during singly linked lists. (15)
6. Compose an algorithm to i) Reverse the elements of a single linked lists. (5) ii) count the number of nodes in a given singly linked list.(5) iii) Searching the element from linked list. (5)
7. Given an list 10,20,30,40 ,generalize the steps to delete a node from the beginning of the linked list, deletion of last node in a list and deletion of middle node in a list.

UNIT II - STACKS, QUEUES

Stack ADT – Operations - Applications – Balancing Symbols -Evaluating arithmetic expressions- Infix to postfix Conversion – Function calls - Queue ADT – Operations -Circular Queue - DeQueue – Applications of queues.

PART – A

1. Point out the advantage of representing stack using a linked list than array.
2. Point out the rules followed during the infix to postfix conversions.
3. Compare the working of stack and queue data structure.
4. Develop an algorithm for inserting a new element into the stack.
5. Define stack and queue and specify its operation.
6. What are the methods to implement stack in C?
7. What are the various Operations performed on the Stack?
8. List any four applications of stack
9. What are the features of stacks?
10. Write a routine for IsEmpty condition of queue.
11. Write the routine to push a element into a stack.
12. What are the methods to implement stack in C?
13. Write the routine to insert a element onto a queue.
14. What are the types of queue?
15. List out the basic operations that can be performed on a stack
16. Define Balancing Symbols.
17. State and explain the different ways of representing expressions.
18. State the rules to be followed during infix to postfix conversions
19. Given the prefix for an expression. Write its postfix: -*-+abc/ef-g/hi
20. Describe how the following "infix" expression is evaluated with the help of stack :
5 * (6 + 2) - 12 / 4
21. Give the postfix and prefix forms of the expression. (i) A + B* (C – D) / (P – R)
(ii) -A+B-C+D
22. Explain the usage of stack in recursive algorithm implementation?
23. Define Function Calls.
24. Mention the advantages of representing stacks using linked lists than arrays
25. Define double ended queue.
26. What are the various operations performed on the Queue?
27. Distinguish between stack and queue.
28. List the applications of a queue.
29. How do you test for an empty queue?
30. What is circular queue?
31. Circular queue is better than standard linear queue, Why
32. Classify the different types of queues.
33. Illustrate the difference between a queues and linked lists with an example

34. Complete a routine to display the contents of queue.
35. Write down the function to insert an element into a queue, in which the queue is implemented
as an array.
36. Analyze and write a routine to check whether the queue is full or empty.
37. For railway reservation the queue data structure is preferred –Justify.
38. Define Dequeue.
39. Define double ended queue.
40. Develop an algorithm for deleting an element in a double ended queue.

PART – B

1. Describe with an example how to evaluate arithmetic expressions using stacks. (13)
2. Explain array based implementation of stacks. (7)
Explain linked list implementation of stacks. (6)
3. i. Describe about stack ADT in detail. (7)
ii. Explain any one application of stack.(6)
4. Explain the following expressions with an example. i.Prefix and infix .(7) ii.Postfix. (6)
5. i. Write an algorithm to convert an infix expression to a postfix expression. Trace the algorithm to convert the infix expression '(a+b)*c/d+e/f' to a postfix expression.(8)
ii. Justify the need for Infix and Postfix expression. (5)
6. i. Give an algorithm for push and pop operations on stack using a linked list.(7)
ii. Discuss about addition and deletion operations performed on a circular queue with
` necessary algorithms. (6)
7. i. Describe the process of postfix expression evaluation with an example. (7)
ii. Describe the process of conversion from infix expression to postfix expression using stack.
(6)
8. i. Write an algorithm that checks if expression is correctly parenthesized using stack and
illustrate with an example. (7)
ii. Write the function to examine whether the stack is full() or empty(). (6)
9. i. Describe about queue ADT in detail. (7)
ii. Explain any one application of queue with suitable example. (6)
10. Briefly describe the operations of queue with examples. (13)
11. Analyze and write an algorithm to implement queue functions using arrays. (13)
12. Develop an algorithm to perform the four operations in a double ended queue that is implemented as an array. (13)
13. Discuss circular queue and its implementation. (13)
14. Illustrate the enqueue and dequeue operations on double ended queues. (13)
15. Explain linear linked implementation of Stack and Queue?
16. (a) Write an ADT to implement stack of size N using an array. The elements in the stack are

to be integers. The operations to be supported are PUSH, POP and DISPLAY. Take into account
the exceptions of stack overflow and stack underflow. (8)
(b) A circular queue has a size of 5 and has 3 elements 10,20 and 40 where F=2 and R=4. After inserting 50 and 60, what is the value of F and R. Trying to insert 30 at this stage what
happens? Delete 2 elements from the queue and insert 70, 80 & 90. Show the sequence of
steps with necessary diagrams with the value of F & R. (8 Marks)
17. Explain in detail about priority queue ADT in detail?
18. i) Trace the algorithm to convert the infix expression "3-(4/2) + (1*5) + 6" to a postfix
expression using stack. (8)
ii) Show the simulation using stack for the following expression to convert infix to postfix :
p*q+ (r-s/t).
19. Explain how to evaluate the following arithmetic expressions using stacks.
i) 6 5 2 3 + 8 * + 3 + * (7)
ii) 2 3 1 * + 9 –
20. Explain in detail about balancing symbols with example.
21. Discuss in detail about function calls.

PART – C

1. Develop and Show the simulation using stack for the following expression :
12 + 3 * 14 – (5 * 16) + 7 . (15)
2. Explain an algorithm to implement the circular queue using arrays. List the applications of
Queues.(15)
3. Assess the difference between double ended queue and circular queue. Show the simulation
`using stack for the following expression to convert infix to postpix : p * q = (r-s / t).(15)
4. Develop an algorithm to explain deQueue and the applications of queues. (15)
5. Develop a C program for linked list implementation of stack.
6. Convert the following infix expression into postfix form:
i) A+B*C+(D*E+F)*G (5)
ii) A+B*((C+D)/F)+G*E (5)
iii)Evaluate the above postfix expression for A=2,B=2,C=1,D=4,E=6,F=4,G=3. (5)
7. A circular queue has a size of 5 and has 3 elements 10,20 and 40 where F=2 and R=4.After
inserting
50 and 60,what is the value of F and R.Trying to insert 30 at this stage what happens? Delete

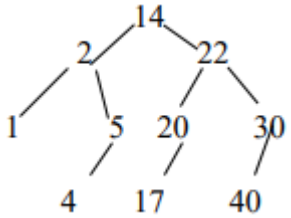
2 elements from the queue and insert 70, 80 & 90.Assess the sequence of steps with necessary
diagrams with the value of F & R. (15)
8. Generalize and develop a function to insert an element into a queue and delete an element
from a queue, in which the queue is implemented as a linked list.

UNIT III - TREES

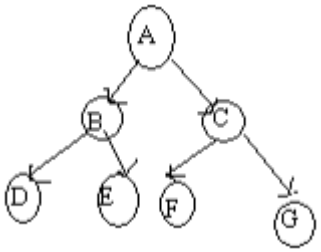
Tree ADT – Tree Traversals - Binary Tree ADT – Expression Trees – Binary Search Tree ADT –- AVL Trees - Priority Queue(Heaps) - Binary Heap.

PART – A

1. Define tree?
2. Define Height of tree?
3. Define Depth of tree?
4. What is the length of the path in a tree?
5. Define sibling?
6. Define forest?
7. Define a complete binary tree
8. If the depth of the binary tree is k, the maximum number of nodes in the binary tree is $2^k - 1$. .Justify
9. For the given binary search tree, if we remove the root and replace it with something from left subtree. What will be the value of the new root? Justify your answer.

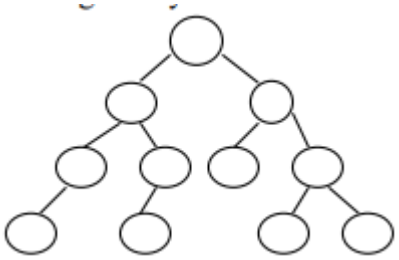


10. Define a fully binary tree.Give an example.
11. What are the two methods of binary tree implementation?
12. Create an expression tree for the expression.a*(b+c)+((d+e*f)*g).
13. Define Binary Search Tree with examples.
14. How does the AVL tree differ from binary search tree?
15. Recommend the result of inserting 3,1,4,6,9,2,5,7 into an initially empty binary search tree
16. What is AVL Tree?
17. What are the various rotations in AVL trees?
18. List the applications of trees.
19. List out few of the Application of tree data-structure?
20. Define a binary tree. Give an example and What are the applications of binary tree?
21. What is meant by binary tree traversal and list out its techniques?
22. Create an expression tree for the expression. (a-b) / ((c*d) + e
23. Convert the infix expression (A-B/C)*(D/E-F) into a postfix. Postfix: ABC/-DE/F-*
24. Give the pre & postfix form of the expression (a + ((b*(c-e))/f)
25. What are the tasks performed during inorder traversal?
26. What are the tasks performed during postorder traversal?
27. Traverse the given tree using Inorder, Preorder and Postorder traversals.



`

28. What are threaded binary trees? Give its advantages.
29. Define balance factor of AVL Tree.
30. How do we calculate the balance factor for each node in a AVL tree?
31. Simulate the result of inserting 3,1,4,6,2,8,9 into an initially empty AVL Tree.
32. Give an example for expression tree.
33. Number the following binary tree to traverse it in i.Preorder ii.Inorder



34. Explain why binary search cannot be performed on a linked list.
35. Illustrate the steps in the construction of a heap of records with the following key values:12,33,67,8,7,80,5,23
36. Analyze the properties of binary heap.
37. Define a heap and show how it can be used to represent a priority queue.
38. Define a priority queue
39. What is the need for Priority queue?
40. What are the applications of priority queue?

PART – B

1. Write an algorithm for preorder, inorder and postorder traversal of a binary tree. (13)
2. Explain the following operations on a binary search tree with suitable algorithms i. Find a node (6)
   ii. Find the minimum and maximum elements of binary search tree (7)
3. i. Write short notes on threaded binary tree (7)
   ii. Describe an iterative algorithm to traverse a tree in preorder (6)
4. Write an algorithm for inserting and deleting a node in a binary search tree. (13)
5. Discuss in detail the various methods in which a binary tree can be represented. Discuss the
   advantage and disadvantage of each method (13)
6. i. What are the steps to convert general tree to binary tree? (6)
7. i.Discuss how to insert an element in a AVL tree, Explain with algorithm. (7)

ii.Explain how deletion can take place in AVL trees with suitable algorithms (6)

8. i.What are AVL trees? Describe the different rotations defined for AVL tree. (7).
   ii.Insert the following elements step by step in sequence into an empty AVL tree15,18,20,21,28,2330,26 (6)
9. Describe the algorithms used to perform single and double rotation on AVL tree.
10. Write short notes on i. Binomial heaps ii. Fibonacci heaps
11. What are threaded binary tree? Write an algorithm for inserting a node in a threaded binary tree
12. Discuss the different traversal technique in binary tree with suitable algorithms and examples? (13)
13. Explain the construction of expression tree with example. (7)
    Give the applications of trees (6)
14. i. Show the result of inserting 15,17,6,19,11,10,13,20,8,14,12 one at a time into an initially empty binary min heap.(7)
    ii.Show the result of performing three delete min operations in the final binary min heap obtained . (6)
15. i.Illustrate How delete operation performed on binary heap? (7)
    ii.Write a suitable operations for procolate up and percolate down operations in a binary heap.(6)
16. Construct an expression tree for the expression (a+b*c) + ((d*e+f)*g). Give the outputs when you apply inorder, preorder and postorder traversals.
17. Create a binary search tree for the following numbers start from an empty binary search tree.
    45,26,10,60,70,30,40 Delete keys 10,60 and 45 one after the other and show the trees at each stage
18. i)Consider the following list of numbers14,15,4,9,7,18,3,5,16,4,20,17,9,14,5 Using that construct a binary search tree. (7)
    ii)Explain the steps to convert general tree to binary tree? (6)
19. Explain in detail about priority queue ADT in detail?
20. Explain how to evaluate the following arithmetic expressions using stacks.
    i) 6 5 2 3 + 8 * + 3 + * (7)
    ii) 2 3 1 * + 9 - (6)
21. i) Trace the algorithm to convert the infix expression "3-(4/2) + (1*5) + 6" to a postfix expression using stack. (8)
    ii) Show the simulation using stack for the following expression to convert infix to postfix :p*q+ (r-s/t). (5)
22. Analyze the implementation of priority queue. (13)
23. Prepare an algorithm to perform the operations in a double ended (13)
24. i) What are AVL trees? Describe the different rotations defined for

AVL tree. (7)

ii) Insert the following elements step by step in sequence into an empty AVL tree 44,30,76,16,39,37.

(6)

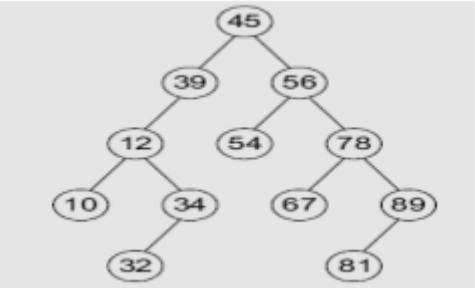
25. Explain the following operations on a binary search tree with suitable algorithms

i) Find a node (7)

ii) Find the minimum and maximum elements of binary search tree. (6)

PART – C

1. Consider the binary search tree given below. Find the result of in-order, pre-order, and post-order traversals. Show the deletion of the root node Insert 11, 22, 33, 44, 55, 66, and 77 in the tree.



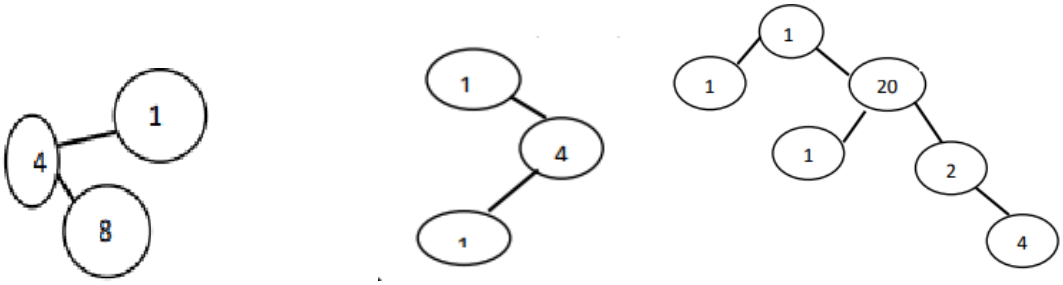
2. **Develop** an algorithm to explain Priority Queue, deQueue and the applications of queues. (15)

3. **Assess** the difference between double ended queue and circular queue.

Show the simulation using stack for the following expression to convert infix to postpix : p * q = (r-s / t). (15)

3. Construct AVL tree for the followings after rotation. (4+4+7)

(i) (ii) (iii)



4. Convert the following infix expression into postfix form:

i) A+B*C+(D*E+F)*G

(5)

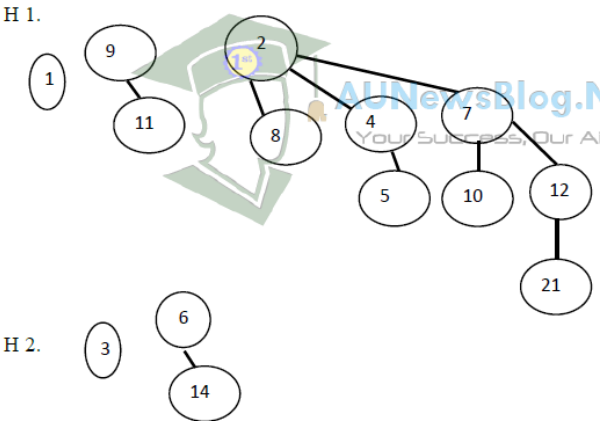
ii) A+B*((C+D)/F)+G*E

(5)

iii)Evaluate the above postfix expression for A=2,B=2,C=1,D=4,E=6,F=4,G=3.

(5)

5. i) Merge the given Binomial heaps. Write procedures for merge operations.



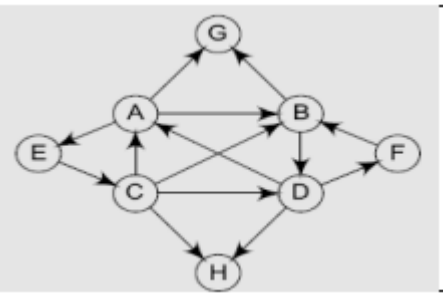
ii)Delete three elements from the merged Binomial Queue. (7)

UNIT IV
MULTIWAY SEARCH TREES AND GRAPHS

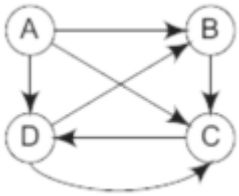
B-Tree – B+ Tree – Graph Definition – Representation of Graphs – Types of Graph - Breadth-first traversal – Depth-first traversal — Bi-connectivity – Euler circuits – Topological Sort – Dijkstra's algorithm – Minimum Spanning Tree – Prim's algorithm – Kruskal's algorithm

PART A

1. Define B tree
2. Define b+ tree.
3. List out the rules of B+ tree
4. Difference between B and B+ tree.
5. How do you calculate the depth of a B-Tree?
6. List out the various operations that can be performed on B-trees
7. List out the properties of B+ -Trees
8. What is a graph and its types?
9. Write the definition of weighted graph?
10. Define adjacency matrix?
11. Define adjacent nodes?
12. Consider the graph given below. Create the adjacency matrix of it



13. Find out the in-degree and out-degree of each node in the given graph



14. What is a directed graph and undirected graph?
15. Create a complete undirected graph having five nodes
16. Given the following adjacency matrix, draw the weighted graph.

$$\begin{pmatrix} 0 & 4 & 0 & 2 & 0 \\ 0 & 0 & 0 & 7 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

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

V(G)=1,2,3,4

E(G) ={ (1,2),(1,3),(3,3),3,4),(4,1)}

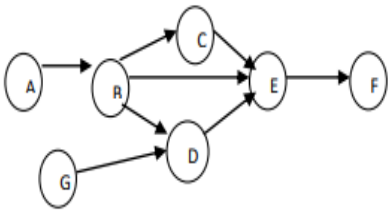
18. Differentiate BFS and DFS.
19. What is a loop?
20. What is a simple graph?
21. What is meant by bi-connected graph.
22. When do you say a graph is bi-connected
23. Give the purpose of Dijikstra's algorithm
24. Differentiate cyclic and acyclic graph
25. Classify strongly connected and weakly connected graph
26. How to find all articulation points in a given graph?
27. Define path in a graph?
28. Define prims algorithm.
29. What is a cycle or a circuit?
30. What is an acyclic graph?
31. What is meant by strongly connected in a graph?
32. When a graph said to be weakly connected
33. Name the different ways of representing a graph? Give examples
34. What is an undirected acyclic graph?
35. What is meant by depth?
36. What are the two traversal strategies used in traversing a graph
37. What is the use of BFS?
38. Advantages and Disadvantages of BFS and DFS.
39. Define the length of the graph.
40. Define Basic Operations of Graph
41. Define minimum spanning tree. Give an example
42. What is topological sort?
43. What is Articulation Points (or Cut Vertices) in a Graph
44. State the principle of Topological sorting.
45. Explain procedure for Depth first search algorithm.
46. Write BFS algorithm
47. Analyze Bi-connectivity.
48. Prove that the number of edges in a complete graph of n vertices in n(n-1)/2
49. In a complete graph with n vertices, show that the number of spanning trees is at least 2 n-1 _ 1

50. Assess the minimum number of spanning tree possible for a complete graph with n vertices.
51. What are Euler circuits?
52. Give two applications of graphs.
53. What is residual graph?
54. Define Kruskals algorithm.

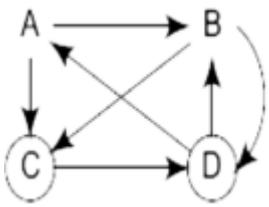
PART B

1. Describe in detail about the following representations of a graph.
   i. Adjacency Matrix (7)
   ii. Adjacency List (6)
2. i. Explain the B+ tree and its properties with an Example (7)
   ii. What are the steps to convert general tree to binary tree? (6)
3. i. Construct B Tree to insert the following key elements(order of the tree is 3)
   5,2,13,3,45,72,4,6,9,22 (7)
   ii. Draw a B Tree of order 6 (6)
4. i) Construct B Tree of order m=5 for the following keys
   1,12,8,2,25,5,14,28,17,7,52,16,48,68,3,26,29,53,55,45 (8)
   ii)Delete the keys 8 and 55.State the rules for deletion. (5)
5. Explain about B Tree with suitable example.
6. Explain about B+ trees with suitable algorithm

7. i. Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on 'D'



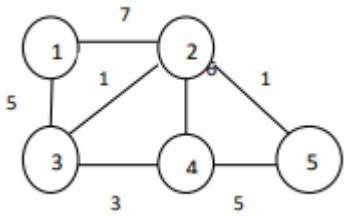
ii. Consider the graph given below and show its adjacency list in the memory



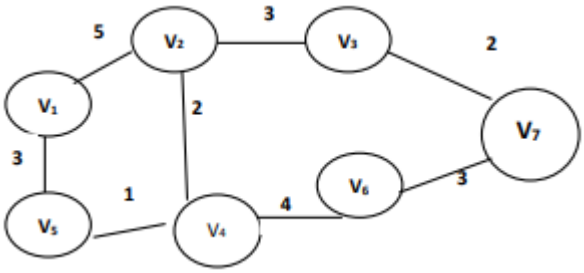
8. i. Explain the topological sorting of a graph G with example. (7)
   ii. Quote the step wise procedure for topological sort (6)
9. Differentiate depth-first search and breadth-first search traversal of a graph with suitable examples. (13)
10. i.Explain with algorithm, How DFS be performed on a undirected graph. (7)
    ii.Show the algorithm for finding connected components of an undirected graph using

DFS, and derive the time complexity of the algorithm. (6)

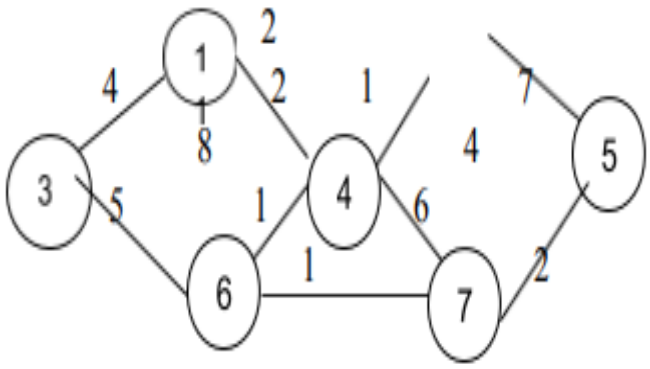
11. i. Discuss an algorithm for Breadth first Search on a graph.(7)
    ii. Give an example based on the algorithm.(6)
12. i. Illustrate Kruskal's algorithm to find the minimum spanning tree of a graph. (7)
    ii.Trace the algorithm for the following graph



13. Compare any two applications of Graph with your own example (13)
14. Describe any one of the shortest path algorithms with suitable example (13)
15. Discuss the prims algorithm for minmum spanning tree.Give an example. (13)
16. i.Write a program to find an Euler circuit in a graph. (7)
    ii.Trace the algorithm for the given graph.(6)



17. Develop an algorithm to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example. (13)
18. Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm.(13)
19. i. Write short notes on Bi-connectivity. (7)
    ii. Express different types of graphs with example. (6)
20. What is topological sort? Write an algorithm to perform topological sort?(8)
21. Explain the various applications of Graphs
22. For the graph given below, construct Prims algorithm



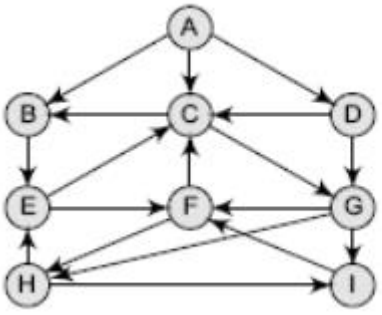
PART C

5. i. Compare B trees with B+ trees. (7)
   ii. Create a B+ tree of order 5 for the following data arriving in sequence:
   90, 27, 7, 9, 18, 21, 3, 4, 16, 11, 21, 72 (8)
6. i. Draw B – Tree pf order m = 5 for the keys {K, O,S,V,MF,B,G,T,U,W} (5)
   ii. Delete the keys K and G in order. (5)
   iii. Justify the number of splits needed for inserts / delete with proper reasons. (5)

7. Given the adjacency matrix of a graph, write a program to calculate the indegree and the out- degree of a node N in the graph. (15)
8. Consider five cities: (1) New Delhi, (2) Mumbai, (3) Chennai, (4) Bangalore, and (5) Kolkata, and a list of flights that connect these cities as shown in the following table.Use the given information to construct a graph.(15)

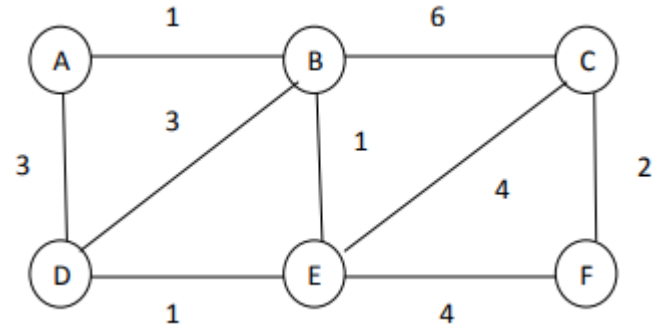
Flight No	Origin	Destination
101	2	3
102	3	2
103	5	3
104	3	4
105	2	5
106	5	2
107	5	1
108	1	4
109	5	4
110	4	5

9. i. How can we efficiently check whether or not a graph is disconnected? (7)
   ii.Describe an algorithm that uses a brute force approach to find all the articulation points in G in O(V (V + E)) time. (8)
10. i. Given a rooted tree, one desires to find the shortest path from the root to a given node v. which algorithm would one use to find this shortest path.(7)
    ii. Write a program to determine whether there is at least one path from the source to the destination. (8)
11. Consider the graph G given below. The adjacency list of G is also given. Evaluate the steps needed to print all the nodes that can be reached from the node H (including H itself).One alternative is to use a depth-first search of G starting at node H.



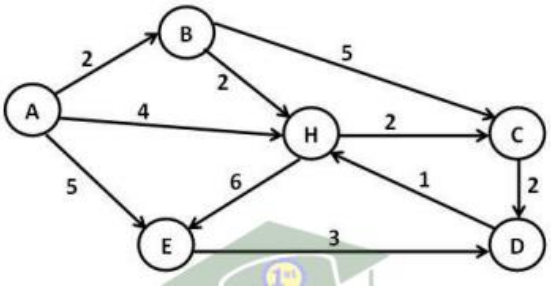
Adjacency lists
A: B, C, D
B: E
C: B, G
D: C, G
E: C, F
F: C, H
G: F, H, I
H: E, I
I: F

12.i) Formulate the minimum spanning tree for the following graph. (8)



ii) Generalize any two applications of depth first search. (7)

13.Using Dijkstra's algorithm to find the shortest path from the source node A.



14.i) Explain weakly connected graph and strongly connected graph with example. (7)

ii) State the various graph traversal algorithm. Explain each in detail. (8)

UNIT V
SEARCHING, SORTING AND HASHING TECHNIQUES

Searching- Linear Search - Binary Search. Sorting - Bubble sort - Selection sort - Insertion sort -Merge sort. Hashing- Hash Functions – Separate Chaining – Open Addressing – Rehashing – Extendible Hashing.

PART A

1. What is hashing.
2. Define extendible hashing and give its significance.
3. Give the fastest searching algorithm
4. What is meant by internal and external sorting? Give any two examples for each type.
5. List the different types of searching
6. Define rehashing.
7. What is meant by internal and external sorting? Give any two examples for each type.
8. Describe the complexity of bubble sort.
9. State the logic of bubble sort algorithm.
10. Name the applications of linear and binary search techniques.
11. When does the Bubble Sort Algorithm stop?
12. Give the time complexities of bubble sort and quick sort
13. Predict the fastest sorting algorithm, justify.
14. What is meant by Sorting?
15. Compare internal and external sorting.
16. Apply insertion sort and sort the following elements 3,1,4,1,5,9,2,6,5
17. Distinguish between linear and binary search technique.
18. Analyze why do we need a hash table as a data structure as compared to any other data structure?
19. Classify the different sorting methods.
20. Develop an algorithm for a linear search.
21. Why bubble sort is called so?
22. What is the output of selection sort after the 2nd iteration given the following sequence?
    16 3 46 9 28 14
23. Point out the advantages of using open addressing
24. How many key comparisons and assignments an insertion sort makes in its worst case?
25. Which sorting algorithm is best if the list is already sorted? Why?
26. Why Shell Sort is known diminishing increment sort?
27. .What is the output of quick sort after the 3rd iteration given the following sequence?
    24 56 47 35 10 90 82 31
28. Which hashing technique is best and illustrate with an example?
29. Summarize the open addressing hashing method with an example.
30. Point out the advantages of using quick sort.
31. Compare the working of linear and binary search techniques.
32. What is divide-and-conquer strategy?

33. Select the best sorting method out of the following - insertion sort, quick sort and merge sort and give justification.
34. Illustrate the time complexity of insertion sort with an example.
35. Identify the advantage of shell sort over insertion sort.
36. What do you mean by hash function?
37. Write the importance of hashing.
38. Give the types of collision resolution
39. Develop a simple algorithm for a linear search.
40. Compare the working of separate chaining and open addressing techniques.
41. Develop an algorithm for a shell sort**.**
42. What are applications of hashing?
43. Write the advantage and disadvantage of separate chaining.
44. What do you mean by Probing?
45. What do you mean by linear probing?
46. What do you mean by primary clustering?
47. What do you mean by quadratic probing?
48. What do you mean by secondary clustering?
49. What do you mean by double hashing?
50. List the limitations of linear probing.
51. Mention one advantage and disadvantage of using quadratic probing.

PART – B

1. Describe how the divide and conquer technique is implemented in binary search. (13)
2. Describe about selection sort with suitable example.
3. Describe the algorithm to sort the following array: 77, 33, 44, 11, 88, 22, 66, 55
   i.Insertion sort (7)
   ii.Shell Sort (6
4. List the different types of hashing techniques and Explain them in detail with an Example. (13)
5. i. Interpret 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. (7)
   ii. Discuss the running time of Divide-and-Conquer Merge sort algorithm. (6)
6. i. Sort the sequence 3, 1, 4, 1, 5, 9, 2, 6, 5 using Insertion sort. (7)
   ii. Describe the routine for insertion sort. (6
7. Write an algorithm to sort a set of 'N' numbers using quick sort. Demonstrate the algorithm for the following set of numbers: 88,11,22,44,66,99,32,67,54,10. (13)
8. Explain the various collision resolution techniques in detail with an example. (13)
9. Write a C program to search a number with the given set of numbers using binary search.
10. Compare the below different Sorting methods and discuss about each method in a very detailed Manner. i.Bucket Sort. (7) ii.Selection Sort .(6)
11. i. Sort the given integers and Explain the intermediate results using shell sort: 35,12,14,9,15,45,32,95,40,5. (7)
    ii. Write and Explain a C code to sort an integer array. (6)
12. i. Create a algorithm to perform a binary Search. (7)

ii. Develop an algorithm for Merge sort with an example.(6)

13.i. Write short notes on Bubble Sort.(5)

ii. Illustrate an algorithm to sort the elements using bubble sort. (8)

14. Describe the following collision resolution techniques in detail with an example.

i.Separate chaining. (7)

ii.Rehashing. (6)

15. i. Explain different hashing technique. (5)

ii. Explain the rehashing technique with suitable example. (8)

16. Describe the open addressing and chaining methods of collusion resolution techniques in hashing. (13)

17.Compare working of binary search and linear search technique with example.

18.Analyze extendible hashing in brief.

19.Explain in detail about separate chaining.

20.Formulate the rehashing technique with suitable example.

21.Prepare an algorithm to sort the elements using radix sort with example.

22.Write an algorithm to implement Bubble sort with suitable example.

23.Explain any two techniques to overcome hash collision.

24.Write an algorithm to implement insertion sort with suitable example.

25.Write an algorithm to implement selection sort with suitable example.

26.Write an algorithm to implement radix sort with suitable example.

27.Write an algorithm for binary search with suitable example.

28.Discuss the common collision resolution strategies used in closed hashing system.

29.what are the advantages and disadvantages of various collision resolution strategies? (6

PART C

1. Develop an algorithm to search a number in a given set of numbers using binary search. Develop and algorithm to explain Extendible Hashing.(15)
2. Explain a C code to sort an integer array using Selection Sort and Radix Sort.(15)
3. Explain an algorithm for Shell Sort and Merge Sort and explain with example.(15)
4. Prepare a quick sort algorithm and explain with suitable example Give its worst case, average case and best case time complexities.(15)
5. Mention the different Sorting methods and Explain about each method in detailed Manner.
6. Sort the sequence 96, 31, 27,42,76,61,10,4 using shell sort and radix sort and prepare the required steps.
7. Given input {4371,1323,6173,4199,4344,9679,1989} and a hash function h(x) =x mod 10. Prepare the resulting for the following:
   i) Separate chaining hash table. (3)
   ii)Open addressing hash table using linear probing. (4)
   iii) Open addressing hah table using quadratic probing. (4)
   iv) Open addressing hash table with second hash h2(x)=7- (x mod 7). (4)
8. i)Write and explain non-recursive algorithm for binary search. (8)