CHAPTER 1

Introduction and Overview

1. Define the following terms: Data, Data structure, Linear data structure, Nonlinear data structure, Field, Record, File and Entity. 1X
2. Why data structure is necessary? 2

CHAPTER 4

Arrays Records and Pointers

1. What is array? Discuss the operations that are normally performed on any linear structure (array). 4
2. What is linear array? How can you found the no. of elements in any linear array? 2
3. Briefly discuss the representation of array in memory? 3
4. How can you traverse an array? 3
5. Briefly discuss the insertion and deletion situation. 3
6. How can you sort the elements of an array? Name some sorting algorithm? 3
7. What is the benefit of binary search over linear search? 2
8. Can we perform binary search on unsorted data? Justify your answer. 2
9. Simulate the binary search algorithm on the following data: 12 34 56 78 89 90 100 103 (suppose we search for item 34). 4
10. What is 2D array? How can you represent 2D array in memory? 4
11. How are the elements of a 2D array stored in the memory? 3
12. How can you find out the no. of elements of a 2D array? Explain with example. 3
13. For row major order find out the address of the element score [10, 2] from a 25X4 matrix array score with base value 200 and w=4. 3
14. For column major order find out the address of the element score [10, 2] from a 25X4 matrix array score with base value 200 and w=4.
15. Maze(2:8, -4:1, 6:10) is a 3D array with base=200, w=4, calculate Maze[5, -1, 8] address in a row major order and column major order.
16. What is pointer and pointer array? 2
17. How a pointer can save memory space to store a 2D array? 4
18. How does a pointer array can save memory when store a variable sized groups of data? Discuss with necessary figures.
19. What is a record? What is the difference between a record and a linear array? 2
20. What is sparse matrix? What is the difference between triangular matrix and Tridiagonal matrix? 3
21. We often store sparse matrix in a 1D array to save spaces. What is the memory saving if we store a sparse matrix in a 1D array rather than a 2D array? 3
22. How can you locate element aij of a sparse matrix from a 1D array? 2

CHAPTER 5

Linked Lists

1. What is a Linked List? Discuss with example. 3
2. What are the advantages of Linked List? 4
3. What are the disadvantages of Linked List? 3
4. How can you represent Linked List in memory? 3
5. Suppose 10 elements are maintained by array and another 10 are by Linked List. Which methods take longer time to access 7th element. Justify your answer. 2
6. What is meant by traversing a Linked List? How can you traverse a Linked List? 3
7. What is the difference between searching a sorted Linked List and an unsorted Linked List? 3
8. What is Garbage collection? When does it take place? 2
9. What is overflow and underflow? How can you handle them? 3
10. One of the advantages of linked list is the ability to insert data into the list easily. Explain with your own words and figures how to insert data at the beginning, after a given node, at the end and to a sorted list. 6
11. Briefly discuss deleting mechanism of an item in the Linked List. 3
12. What is Header Linked List? Explain. 3
13. What is two way lists? Why it is important? Explain with Schematic diagram. 3
14. What is linked list? How are linked lists more efficient than arrays? 4
15. Explain the scenarios where you can use linked lists and arrays. 3

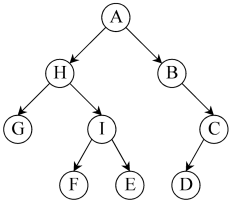
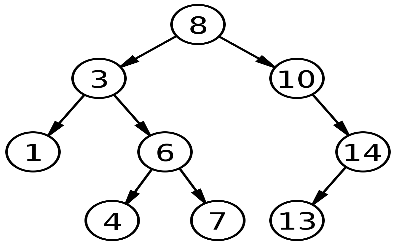
CHAPTER 6

Stacks Queue and Recursion

1. What is stack? What are the operations on stack? Explain with example. 4
2. Write the steps involved in the insertion and deletion of an element in the stack. 3
3. Discuss the array representation mechanism of stack. 3
4. What is polish notation? What are the benefits of polish notation? 2
5. Convert the following infix expression to its equivalent prefix and postfix expression 2X
6. Evaluate the following postfix expression (Manually). 2
7. Simulate the postfix expression evaluation algorithm using 12, 6, /, 6, 2, +, \*, 12, 4, /, - by showing Stack’s contents as each element is scanned. 3
8. Simulate the infix to postfix transformation algorithm for Q: A + (B \* C - (D / E ↑ F) \* G) \* H by showing the stack’s contents as each element is scanned. 4
9. What is recursion? Explain the use of recursion. 3
10. What is Queue? What are the operations on queue? Explain with example. 4
11. How can you represent queue in computer? 3
12. Discuss deque. 3
13. What is priority queue? Why it is important? 2
14. What is a Queue, how it is different from the stack and how is it implemented? 4
15. What are the drawbacks of array implementation of Queue? 2
16. Write in brief the function (i) push(), (ii) pop(), (iii) qinsert() and (iv) qdelete(). 2

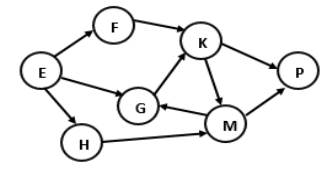
CHAPTER 7

Tree

1. What is binary tree? Discuss with example. 4
2. Illustrate similar and copies with example. 3
3. Define the following terms: siblings, successor, ancestor, depth, level. 1X
4. What is complete binary tree? What is the parent child relationship? 2
5. For 1089 nodes, find out the depth of the tree. 2
6. What is extended binary tree? 2
7. Discuss the linked representation of binary tree in memory. 3
8. Discuss the array representation of binary tree in memory. 3
9. Discuss the steps of preorder, inorder and postorder traversal of binary tree. 3
10. For \*+a-bc/-de-+fgh draw the tree and perform inorder and postorder traversal. 3
11. Simulate the preorder, inorder and postorder traversal algorithm for the following tree. 6X
12. Tree traversal (also known as walking the tree) refers to the process of visiting each node exactly once. Simulate the preorder traversal algorithm for the following tree.
13. What is binary search tree? Why binary search tree? 3
14. Simulate the binary search tree algorithm for the following tree. 4
15. How can you search ’5’ in the following binary search tree? However if ‘5’ is not in the tree just insert to its appropriate place and show the resultant tree.
16. Suppose the following six numbers are inserted in order into an empty binary search tree. 40, 60, 50, 33, 55, 11. Show the tree as each number is inserted into a binary search tree. 3
17. What is the difference between maxheap and minheap? 2
18. Simulate the maxheap algorithm for the following values: 67, 29, 90, 48, 12, 34, 90, 9, and 12. 5

CHAPTER 8

Graph

1. Define the following terms: Graph, End Point, Adjacent Nodes/Neighbor, Degree of a node, Isolated node, Path, Closed Path, Simple Path, Cycle, Connected Graph, Complete Graph, Tree Graph/Free Graph, Labeled Graph, Weighted Graph, Multi Graph, Finite Graph.
2. What is Directed Graph? Explain. 3
3. Define the following terms for Directed Graph: Sink, Reachable, Connected/Strongly Connected, Unilaterally Connected, Parallel Edge, Loop, Path, Indegree, Outdegree, and Simple Graph.
4. Discuss the sequential Representation of Graph with example. 5
5. What is adjacency matrix? How is it formed? 2
6. Consider the following adjacency matrix below:
7. Now find out A2, A3, A4, B4 and from that make the path matrix and tell whether this is strongly connected or not. 5
8. Use the Warshall’s algorithm to find the path matrix of the adjacency matrix given below. 4
9. Use the Warshall’s algorithm to find the shortest path matrix of the weighted matrix given below. 5
10. Discuss the linked representation of Graph with example. 5
11. How many ways a graph G can be traversed? What is the significance of the STATUS field? 2
12. Consider the following graph that represents the daily flights between cities of some airline, and find the path if someone want to fly from city E to city P with minimum no. of stops.
13. Consider the adjacency list of the Graph G in the following table. Draw the graph and find out the path from A to F with minimum number of nodes along that path using Breadth First Search. 5

|  |  |  |  |
| --- | --- | --- | --- |
| Node | Adjacency | Node | Adjacency |
| A | E,G | E | H |
| B | C | F | A, B |
| C | F | G | B, C, E |
| D | C | H | D |

1. Consider the adjacency list of the Graph G in the following table. Find the nodes that are reachable from node C using Depth First Search. 4

|  |  |  |  |
| --- | --- | --- | --- |
| Node | Adjacency | Node | Adjacency |
| A | G, E | E | C |
| B | C | F | A, B |
| C | F | G | B, C, E |
| D | C | H | D |

**Huffman codes**

1. Suppose that we want to encode a message constructed from the symbols A, B, C, D, E, F and G using a fixed-length code. How many bits are required to encode the message FDEGAACAAGAAFABA? 2
2. What are the drawbacks of fixed-length codes? What are the potential solution? 2
3. What are the advantages of variable-length codes? What is its potential problem? 2
4. Discuss the Huffman coding tree with example. 4
5. Build the Huffman coding tree for the message ‘message passage’. 4
6. Build the Huffman coding tree for the message ‘science engineering’. 4

**Sorting Algorithms**

1. All Sorting algorithms.