Class: SY BSC IT

Subject: Data Structures

UNIT 1

1. Define Data Structures. What are different types of data structure?
2. List and explain different operations that can be performed on a data structure.
3. Define Algorithms. Explain its characteristics.
4. Write a short note on Asymptotic Notations.
5. Explain Array with an example.
6. Explain Memory representation of one dimensional array.
7. Write a program to traverse elements of an array and print them.
8. Write a program to insert an element in an array
9. Write a program to delete an element in an array.
10. Write a program to sort elements of an array.
11. Write a program to split an array into two.
12. Write a program to merge two arrays.
13. Write a program to search for an element in an array.
14. Define a two dimensional array with an example.
15. Explain Memory representation of a 2D array.
16. Write a short note on a sparse array.
17. List advantages and disadvantages of an array.
18. Find the number of elements of the following arrays X[7:19], Y [—2:7]
    a. If the base address of array X is 1500 and each element takes 4 memory cells then find the location of the 5th element of the array.
    b. If the base address of array Y is 1000 and each element of array takes 10 memory cells then find the location of the 3rd element of the array.
19. Consider a two dimensional array £> [5: 1, —3:6]. If the base address of D is 1536 and each element takes 2 memory cells then find the address of d 60 element assuming that
    a. Array D is stored in column major
    b. Array D is stored in row major
20. Explain the working of sorting algorithm on the following elements.
    30 20 50 40 10 70 60 80

UNIT 2

1. List and explain advantages of linked list over array.
2. Explain memory allocation and de-allocation functions.
3. Write a program to insert a node at the start of a linked list.
4. Write a program to insert a node at the last of a linked list.
5. Write a program to insert an intermediate node in a linked list.
6. Write a program to search a value in the linked list.
7. Write a program to delete a node from a linked list (start, intermediate, and end).
8. Write a program to copy a linked list into another.
9. Write a program to merge two linked list.
10. Write a program to split a linked list into two.
11. Write a program to reverse a linked list.
12. Write a short note on applications of a circular linked list.
13. Write a program to insert a node at the start of a circular linked list.
14. Write a program to insert a node at the last of a circular linked list.
15. Write a program to delete a node from a circular linked list (start, end).
16. Write a program to insert a node into a doubly linked list (start, intermediate, and end).
17. Write a program to delete a node from a doubly linked list (start, intermediate, and end).
18. Short note on header linked list.
19. Explain representation of polynomials using linked list.

UNIT 3

1. Define stack. What are operations performed on stack?
2. Explain array representation of a stack.
3. Explain linked list representation of a stack.
4. Explain Push operation with an example.
5. Explain pop operation with an example.
6. Explain peek operation with an example.
7. Write a program for matching parenthesis.
8. Write an algorithm to convert an infix expression to postfix expression using stack.
9. Write an algorithm to evaluate a postfix expression.
10. Write a short note on recursion.
11. Short note on Queue.

12. Explain operations performed on array representation of queue(enqueue, dequeue).
13. Explain operations performed on linked list representation of queue(enqueue, dequeue).
14. Explain operations performed on circular queue(enqueue, dequeue).
15. Short note deque.
16. Short note on application of priority queue.

UNIT 4

1. Explain selection sort algorithm with an example.
2. Explain insertion sort algorithm with an example.
3. Explain bubble sort algorithm with an example.
4. Reconstruct the binary tree from traversal.
5. Differentiate between sequential and binary search.
6. Write algorithm for sequential search and explain with an example.
7. Write algorithm for binary search and explain with an example
8. Define binary search tree. Write an algorithm to find the position of a given element 'Item' and its parent in a binary search tree.
9. Define AVL tree. How is balancing performed on AVL tree?
10. Write a short note red black tree.
11. Explain Huffman algorithm.
12. Define Binary Tree. Explain its properties.
13. Write a short note B-tree.
14. Write a short note 2-3 tree.
15. Explain heap sort algorithm.

UNIT 5

1. Describe hashing, hash table and hashing function with example.
2. Describe a linear probing with an example.
3. Explain Adjacency matrix.
4. Write a short note on Warshall's Algorithm.
5. Explain address calculation techniques.
6. Describe a quadratic probing with an example.
7. Define the following terms

Graph, Directed graph, Weighted graph, Vertex, path

8. Write a short note on applications of graph.
9. Explain rehashing technique in detail.
10. Explain the difference between linear and quadratic probing with example.
11. Define the following terms: Undirected graph, Edge, Indegree, Outdegree, Cycle
12. Write a short note on linked list and adjacency matrix representation of a graph.
13. Explain Breadth First Search and Depth First Search traversal technique of a graph.
14. Define Spanning tree. Explain prim's algorithm with an example.
15. Explain Kruskal's algorithm for a minimum spanning tree with an example.
16. Traverse the tree using dept first and breadth first search.