

# DS (CHAPTER WISE IMP Q'S)

---

## UNIT 1 - INTRODUCTION TO DS

1. **Converting infix to postfix**
2. Evaluating given postfix/prefix using Stack
3. Programme to implement stack/Queue
4. **Tower of Hanoi (programme & algorithm)**
5. Define Stack/Queue and their applications
6. Parenthesis matching using Stack

---

## UNIT 2 -Tree DS

1. **Differentiate between using Linked List and Array for implementing Tree DS.**
  2. **Inorder, Preorder, Postorder:**
    - Algorithms (or)
    - Programme (or)
    - Writing the order of given tree
  3. **Defining and constructing Min (or) Max Heap with given data**
  4. Representing given Arithmetic expression in binary tree format
  5. Algorithm for Heap Sort.
-

## UNIT 3 - Advance Trees

### 1. Constructing:

- AVL
- B-Tree
- BST(insertion & deletion)
- Red-black tree

### 2. Write short note on:

- B-tree
- Red-Black tree
- Splay tree

### 3. Properties of Red-Black tree

### 4. Algorithm/explanation for:

- B-Tree
- AVL rotations, insertion, deletion
- BST Insertion/deletion

*IgniteXT*

---

## UNIT 4 - Graphs

### 1. Describe in detail about the following representations of a graph.

- i) Adjacency Matrix
- ii) Adjacency List

### 2. Explain and display BFS and DFS for given graph

### 3. Elaborate Spanning Trees and its properties

### 4. Explain and implement Dijkstra Algorithm .(for given graph)

### 5. Find minimum spanning tree using

- Prims algorithm
- Kruskal's algorithm

---

## UNIT 5 - Hashing

1. (i) Define hash function and discuss the applications of hashing.

(ii) Summarize Separate Chaining Concept.

### 2. Open addressing hash table using

- linear probing.
- Quadratic probing
- Double hashing

3.) i) Write a program for implementation of Dictionaries

ii) Categorize Rehashing & Extendible Hashing

### 4. Construct hash table by using Extendible Hashing with the given data

5. What is collision? Explain different collision resolution techniques with example?

---

*IgniteXT*