B. TECH. INFORMATION TECHNOLOGY

UIT2504 ARTIFICIAL INTELLIGENCE

Rohith Varshighan S

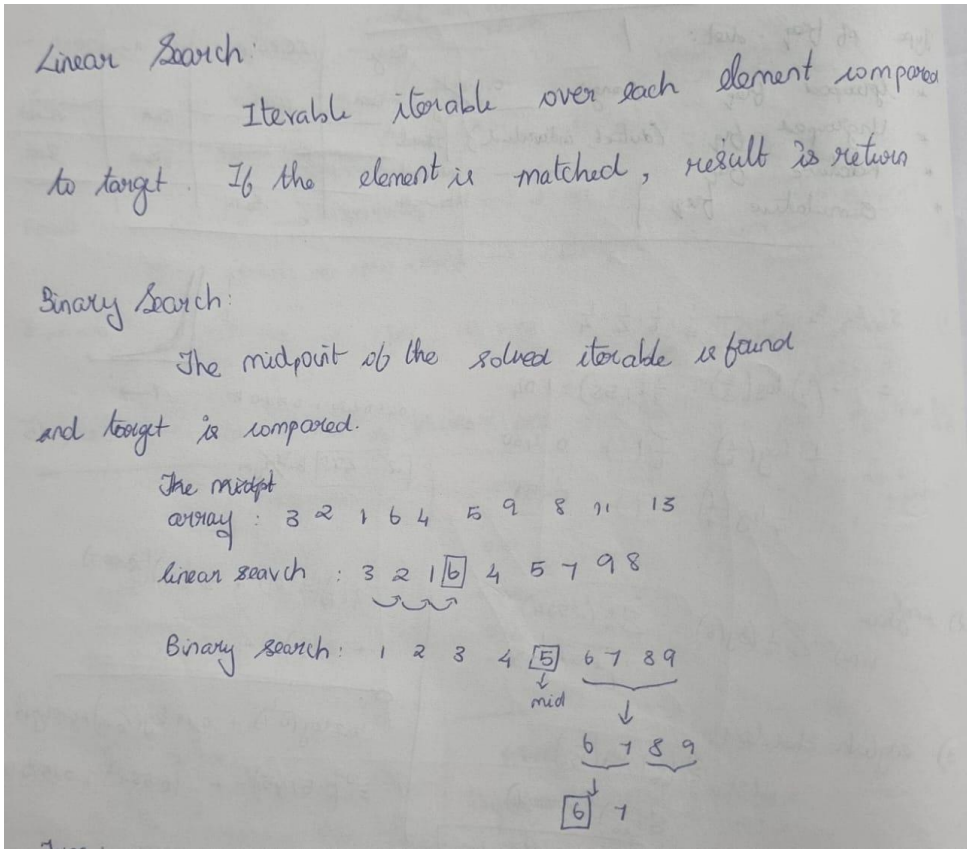
3122235002101

IT – B

---

1. Linear & Binary Search

Linear search is a simple searching technique that checks each element in a list or array sequentially until the target element is found or the list ends. It has a time complexity of **O(n)** and works on unsorted data. **Binary search**, on the other hand, is a more efficient algorithm with a time complexity of **O(log n)** but requires the data to be sorted. It repeatedly divides the search interval into half by comparing the target element with the middle element of the list, discarding one half of the data in each step until the target is found or the search space becomes empty.



---

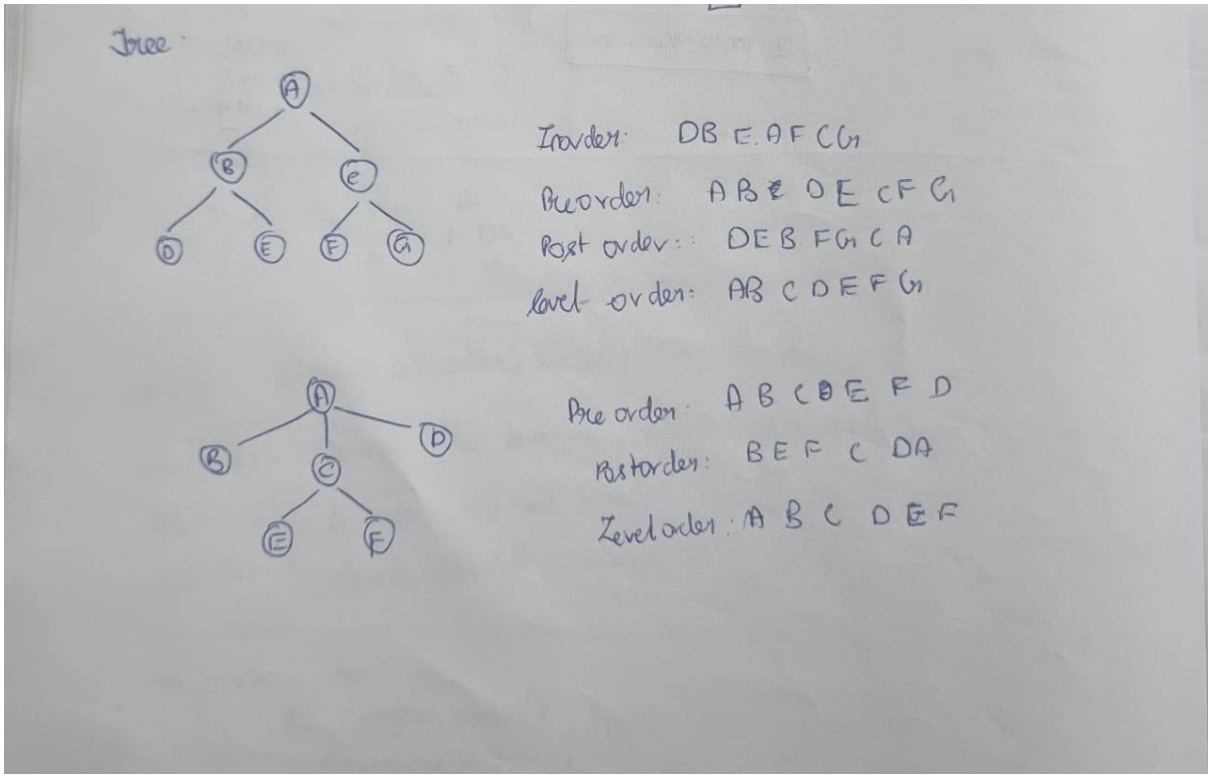
2. Binary Tree Traversal

Binary tree traversal refers to the process of visiting all the nodes in a binary tree systematically. The three main types are **Inorder (Left → Root → Right)**, **Preorder (Root → Left → Right)**, and **Postorder (Left → Right → Root)**. Each traversal order is used for specific applications such as expression tree evaluation or hierarchical data representation. Additionally, **Level-order traversal** (using a queue) visits nodes level by level from top to bottom and left to right, commonly implemented using a **Breadth-First Search (BFS)** approach

---

3. General Tree Traversal

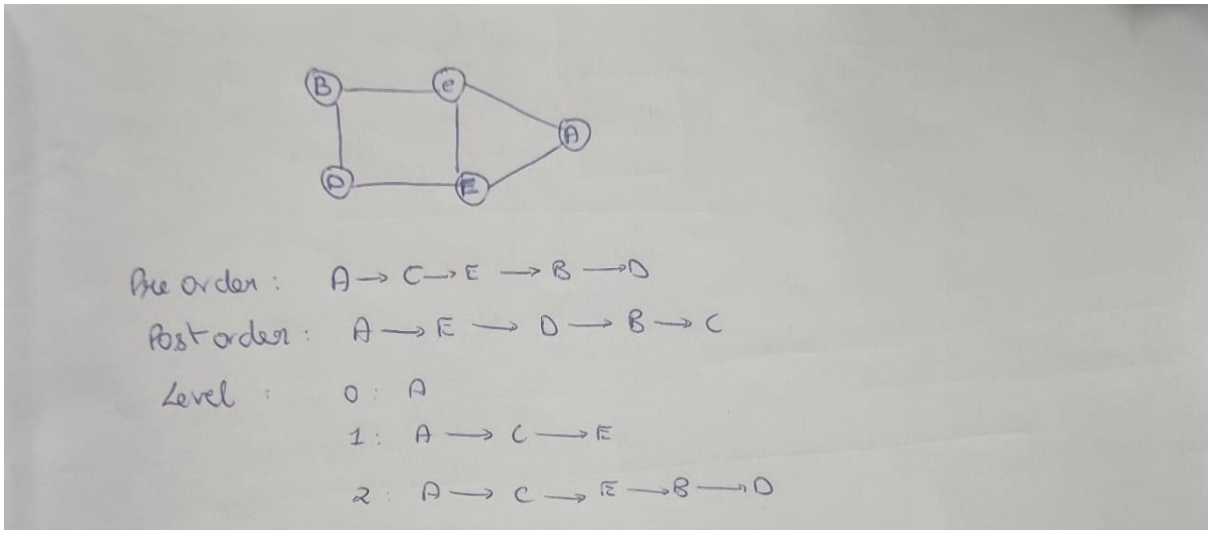
General trees (where nodes can have multiple children) can be traversed in various ways. The most common approaches are **Depth-First Search (DFS)** and **Breadth-First Search (BFS)**. DFS can be implemented in three forms—**Preorder**, **Postorder**, and **Inorder** (for binary-like general trees). BFS uses a queue to traverse the tree level by level. General tree traversal is widely used in file systems, parsing, and hierarchical data structures like XML or JSON.



---

4. Graph Traversal

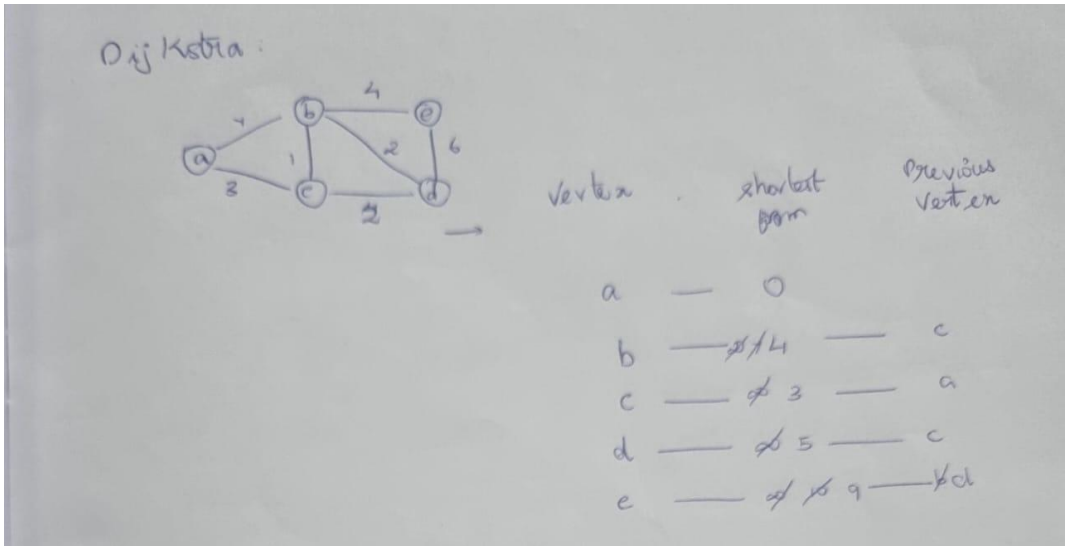
Graph traversal is the process of visiting nodes (vertices) of a graph in a systematic way. Two main techniques are **Depth-First Search (DFS)**, which explores as far as possible along each branch before backtracking, and **Breadth-First Search (BFS)**, which explores all neighbors of a node before moving to the next level. Graph traversal is essential for tasks like finding connected components, pathfinding, and cycle detection. Depending on whether the graph is directed or undirected, the traversal rules may slightly differ.



---

5. Shortest Path Algorithms

Shortest path algorithms are used to find the minimum distance or cost between nodes in a graph. The most common algorithms include **Dijkstra's algorithm** (for graphs with non-negative weights), **Bellman-Ford algorithm** (handles negative weights), and **Floyd-Warshall algorithm** (finds shortest paths between all pairs of nodes). These algorithms are widely applied in network routing, GPS navigation, and optimization problems.



---

6. Minimum Spanning Tree (MST)

A Minimum Spanning Tree of a weighted, connected, undirected graph is a subset of edges that connects all vertices with the minimum possible total edge weight, without forming cycles. Two well-known algorithms for finding an MST are **Kruskal's algorithm** (which sorts edges and adds them while avoiding cycles) and **Prim's algorithm** (which grows the MST starting from an arbitrary node). MSTs are widely used in designing network infrastructures like roads, electrical grids, and communication networks.