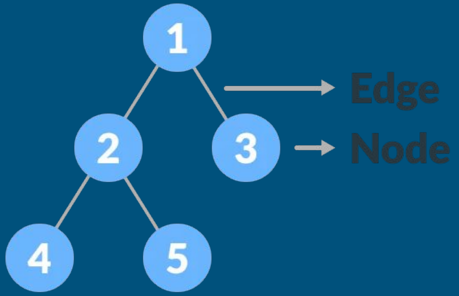
Understanding Tree Data Structure

Cracking Coding Interview @ Ostad
- Partharaj Deb -

Tree Data Structure

A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges.

- **Nodes** in a tree represent entities
- **Edges** represent relationships between entities.

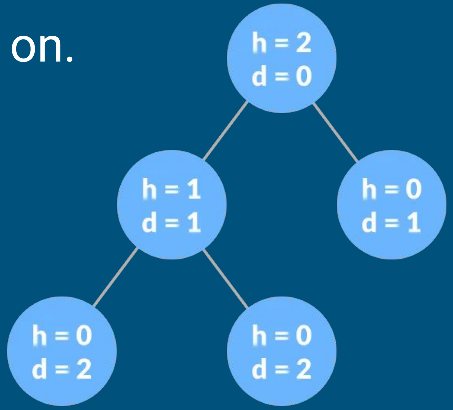


Basic Components

- Node: Fundamental building block containing data.
- Edge: Connection between nodes, depicting a relationship.
- Root: Topmost node in the tree.
- Leaf: Node with no children.

Terminology

- **Parent**: A node with child nodes.
- **Child**: Nodes connected to a parent.
- **Siblings**: Nodes sharing the same parent.
- **Ancestor**: A node's parent, grandparent, and so on.
- **Descendant**: A node's children, grandchildren, and so on.
- **Height of a Node**
- **Depth of a Node**
- **Height of a Tree**
- **Degree of a Node**
- **Forest**



Common Types of Tree

- Binary Tree
- Ternary Tree
- N-ary/Generic Tree
- Binary Search Tree
- Balanced Tree
- B-Tree
- AVL Tree

Basic Operations

- Create: create a tree in the data structure.
- Insert: Inserts data in a tree.
- Search: Searches specific data in a tree to check whether it is present or not.
- Traversal:
    - Depth-First Traversal
        - In-Order: Left subtree, node, right subtree.
        - Pre-Order: Node, left subtree, right subtree.
        - Post-Order: Left subtree, right subtree, node.
    - Breadth-First Traversal

Tree Representation

1. 1D Array Representation
2. 2D Array (Adjacency Matrix) Representation
3. Adjacency List Representation
4. Class/Object Representation

Applications of Trees

- File Systems:
  - Representing hierarchical directory structures.
- Database Indexing:
  - Implementing indexing structures for efficient search operations.
- Expression Trees:
  - Representing mathematical expressions for efficient evaluation.
- Abstract Syntax Trees (AST):
  - Used in compilers for syntax analysis and representing program structure.
- Network Routing Algorithms:
  - Representing network routing tables efficiently.

Q & A