**Day15 docs: 5th july 2025**

**Task -6:**

## **Applications of Trees:**

### 1. Hierarchical Data Representation:

* Trees are perfect for showing hierarchy — one item at the top (root) and others below it (children).
* Examples:  
  + File system (folders and subfolders)
  + Organization charts (CEO → Managers → Employees)
  + Family trees

### 2. Binary Search Trees (BST):

* Used for fast searching, insertion, and deletion.
* Examples:  
  + Searching in databases
  + Auto-complete suggestions
  + Symbol tables in compilers

### 3. Heaps (Special Trees):

* Used to implement priority queues.
* Examples:  
  + Task scheduling in operating systems
  + Dijkstra’s shortest path algorithm
  + Heap sort algorithm

### 4. Tries (Prefix Trees):

* Efficient for searching strings and prefixes.
* Examples:  
  + Auto-complete in search engines
  + Spell checkers
  + IP routing

### 5. Parse Trees / Syntax Trees:

* Used in compilers and interpreters to parse expressions and code.
* Examples:  
  + Converting code into machine language
  + Analyzing expressions like 3 + (4 \* 5)

### 6. Decision Trees:

* Used in machine learning and AI.
* Examples:  
  + Decision-making models (e.g., loan approval)
  + Classification problems (e.g., spam or not spam)

### 7. XML/HTML Document Structure:

* Web documents are structured like trees.
* Examples:  
  + DOM (Document Object Model)
  + Browsers parse HTML using tree structures

### 8. Databases (B-Trees & B+ Trees):

Used in database indexing for fast data retrieval.

Examples:

* MySQL, Oracle, MongoDB use B-trees internally

**Task : 7**

**Different Types of Binary trees**

| **Tree Type** | **Key Feature** |
| --- | --- |
| Full | Each node has 0 or 2 children |
| Complete | All levels filled, last level left to right |
| Perfect | Full + all leaf nodes at same level |
| Balanced | Height difference between subtrees ≤ 1 |
| Degenerate/Skewed | All nodes have one child |
| BST | Left < root < right |
| AVL | Self-balancing BST |
| Red-Black | Self-balancing BST with coloring rules |
| Threaded | Faster traversal using extra pointers |
| Expression Tree | Represents mathematical expressions |

**Task - 8:**

**Different Applications of Graphs:**

## **1. Social Media (Like Facebook, Instagram):**

* People are dots (nodes).
* Friendships or follows are lines (edges).
* Use: To suggest friends, show mutual friends, or form groups.

## **2. Maps & GPS (Like Google Maps):**

* Cities or places are nodes.
* Roads between them are edges.

Use: To find the shortest route from one place to another.

## **3. Computer Networks:**

* Computers or devices are nodes.
* Cables or wireless connections are edges.
* Use: To send data efficiently from one device to another.

## **4. Recommendation Systems (Like Netflix or Amazon):**

* People and movies/products are nodes.
* If someone watches a movie or buys a product, it creates an edge.
* Use: To recommend what to watch or buy next.

## **5. Search Engines (Like Google):**

* Websites are nodes.
* Links between them are edges.
* Use: To decide which website is more important or popular.

## **6. Artificial Intelligence (AI):**

* Ideas or objects are nodes.
* Their relationships are edges.
* Use: To make smart systems that can learn and think like a human.

**Task 9:**  
 **Different Types of Graphs are as follows:**

## **1. Undirected Graph:**

* The **edges (lines)** don’t have a direction.
* A connection goes **both ways**.
* Example: Friendship — If A is a friend of B, B is also a friend of A.

## **2. Directed Graph (Digraph):**

* The edges have an arrow (direction).
* A connection goes only one way.
* Example: Instagram — If A follows B, B may not follow A back.

## **3. Cyclic Graph:**

* A graph that has at least one cycle.
* A cycle means you can start at one node and come back to it by following the edges.
* Example: Road loops in a city.

## **4. Acyclic Graph:**

* No cycles — you can’t return to a node once you leave.
* Example: Task scheduling (Task A → Task B → Task C).

## **5. Weighted Graph:**

* Each edge has a value or weight.
* Example: Map — the weight could be distance between cities.

## **6. Unweighted Graph:**

* Edges do not have values.
* Just shows whether nodes are connected or not.

## **7. Connected Graph:**

* All nodes are connected, either directly or indirectly.
* Example: Every person in a small group can reach every other person.

## **8. Disconnected Graph:**

* Some nodes are not reachable from others.
* Example: Some people in a social app don’t know each other at all.

## **9. Complete Graph:**

* Every node is connected to every other node.
* Example: A small WhatsApp group where everyone messages everyone.