Perfect 👌 Let’s build a Tree Traversal Cheat Sheet for interviews.

This will help you quickly decide BFS vs DFS when you see a tree problem.

🌳 Tree Problems Cheat Sheet: BFS vs DFS

🔷 BFS (Level-Order) – Use when the problem is level-based or “view”-based

👉 BFS processes level by level, so it’s natural for problems about views, distances, or shortest paths.

Common BFS problems:

Right Side View → take the last node of each level.

Left Side View → take the first node of each level.

Zigzag / Level Order Traversal → read level nodes in specific order.

Connect Next Pointers → link children in the same level.

Top / Bottom View → use BFS + horizontal distance map.

Minimum Depth of Binary Tree → BFS gives shortest path to a leaf.

Nodes at Distance K from Target → BFS works well from target outward.

Vertical Order Traversal → BFS with column tracking.

Pattern:

Use a queue.

Process all nodes at one level (loop size = queue length).

Collect the node(s) you need from each level.

🔷 DFS (Preorder / Inorder / Postorder) – Use when problem is recursive or structural

👉 DFS naturally fits recursion and problems about structure, depth, or path sums.

Common DFS problems:

Max Depth of Binary Tree → recursion depth = answer.

Path Sum / All Root-to-Leaf Paths → carry running sum/path.

Diameter of Binary Tree → compute left+right depths in recursion.

Lowest Common Ancestor → recurse into left/right.

Validate BST → recurse with min/max bounds.

Invert Binary Tree → recurse and swap children.

Serialize/Deserialize Binary Tree → DFS preorder works well.

Count Good Nodes / Uni-value Subtrees → check recursively.

Pattern:

Use recursion (or stack if iterative).

Track depth, sum, or path as parameters.

Usually O(N) with recursion depth = tree height.

🔹 Problems solvable by both BFS & DFS

Sometimes either works, but one is more natural:

Right Side View → BFS (last in level) or DFS (right-first preorder).

Minimum Depth → BFS (first leaf found) or DFS (track depth, but less efficient).

Vertical Order Traversal → BFS is common, DFS also possible with column index.

⚡ Interview Tip

When you see a problem:

Mentions "view", level, shortest distance → think BFS.

Mentions path, sum, max/min, recursion → think DFS.

👉 Quick memory hook:

BFS → What I see, level by level.

DFS → What I know, path by path.