Binary Tree Revision Guide (Striver Patterns)

Day 1 — Traversals (DFS + BFS Patterns)

- Recursive Traversals: Preorder (Root-Left-Right), Inorder (Left-Root-Right), Postorder (Left-Right-Root).
- Iterative Traversals: Use stack; for preorder push Right first, then Left.
- Level Order (BFS): Queue-based; process per level.
- Morris Traversals: O(1) space traversal using temporary right links.

Day 2 — Depth, Height & Structural Problems

- Maximum Depth: max(left, right) + 1.
- Identical Trees: Recursive structure and value check.
- Symmetric Tree: Compare mirror nodes.
- Balanced Tree: Compute height bottom-up; return -1 if unbalanced.
- Diameter: Longest path = left height + right height.

Day 3 — Path & Sum Problems

- Max Path Sum: Global max = left_gain + right_gain + node.val.
- Root-to-Leaf Path Sum: Backtrack if sum matches target at leaf.
- Path Sum II: DFS + backtracking to print paths.
- Path Sum III: Prefix-sum + hashmap technique.

Day 4 — Construction & Serialization

- Build from Traversals: Preorder+Inorder or Postorder+Inorder.
- Serialize/Deserialize: Use BFS with null markers.
- Construct from String: Parse using stack (optional).

Day 5 — Views & Width Problems

- Left/Right View: First/Last node per level.
- Top/Bottom View: BFS with horizontal distance mapping.
- Vertical Order: Group by horizontal distance and level.

- Zigzag Traversal: Alternate insertion direction per level.
- Boundary Traversal: Left boundary, leaves, right boundary reversed.

Day 6 — Ancestors, Distances & Counts

- Lowest Common Ancestor: Return node if both sides give non-null.
- Nodes at Distance K: Map parents, BFS outward from target.
- Minimum Time to Burn Tree: Multi-source BFS pattern.
- Count Nodes (Complete Tree): If left height == right height, use formula (2^h - 1).
- Maximum Width: Track level indices like heap positions.

Day 7 — Full Revision Practice

- Mix 10–12 problems across patterns.
- Identify pattern before solving.
- Explain recursion & base cases aloud.
- Use summary table for last-minute glance.

Quick Pattern Summary

- DFS Recursion: Pre/In/Postorder, Depth, Diameter — Recurse left/right.
- BFS Queue: Level, Width, Views — Use queue with level tracking.
- Parent Mapping: For K-distance and Burn Tree problems.
- DP on Trees: Diameter, Max Path Sum, Balanced Tree.
- Hashing + DFS: Path Sum III with prefix sum map.
- Threaded Tree: Morris Traversals (O(1) space).
- Divide & Conquer: LCA, Construct Tree problems.