Same Tree

Key Idea:

Two trees are the same if:

1. Both nodes are null → true.
2. One is null and the other is not → false.
3. Both are non-null:
   - Values must be equal.
   - Left subtrees must be the same.
   - Right subtrees must be the same.

This naturally leads to a recursive solution.

Recursive Solution:

Algorithm:

1. If both p and q are null, return true.
2. If one is null, return false.
3. If p→val != q→val, return false.
4. Recursively check:
   - isSameTree(p→left, q→left).
   - isSameTree(p→right, q→right).

Iterative Approach (Optional):

Use a queue (BFS) or stack (DFS) to compare nodes level by level:

- Push pairs of nodes (p, q) into the queue.
- At each step, compare values and enqueue children pairs.

Complexity:

- Time: O(N) – N is total number of nodes (visit all nodes).
- Space: O(H) for recursion (H = tree height) or O(N) for iterative queue.