

## Construct Binary Tree from Preorder and Inorder Traversal

### Key Observations:

1. Preorder traversal: [Root, Left subtree, Right subtree]
2. Inorder traversal: [Left subtree, Root, Right subtree]
3. First element of preorder is always the root.
4. Locate this root in inorder to:
  - Identify left subtree elements (before root in inorder).
  - Identify right subtree elements (after root in inorder).

### Approach:

- Use recursion with indices (avoid slicing arrays for  $O(1)$  extra space).
- Maintain:
  - preIndex (current root index in preorder)
  - Hash map in Map to store value  $\rightarrow$  index mapping for inorder.

### Steps:

1. Create a map from inorder for  $O(1)$  root index lookup.
2. Recursive function buildTreeHelper(preorder, inStart, inEnd):
  - If inStart > inEnd, return null.
  - Create root from preorder[preIndex], increment preIndex.
  - Find root's index in inorder from inMap.
  - Recursively build left and right subtrees using boundaries.

### Complexity:

- Time:  $O(N)$  - each node visited once, hashmap lookup  $O(1)$
- Space:  $O(N)$  - hash map + recursion stack (worst case skewed tree).