Construct Tree from given Inorder and Preorder traversals

Algorithm: buildTree()

1. Pick an element from Preorder. Increment a Preorder Index Variable (preIndex in below code) to pick the next element in the next recursive call.
2. Create a new tree node tNode with the data as the picked element.
3. Find the picked element's index in Inorder. Let the index be inIndex.
4. Call buildTree for elements before inIndex and make the built tree as a left subtree of tNode.
5. Call buildTree for elements after inIndex and make the built tree as a right subtree of tNode.
6. return tNode.

Implementation in Python

```
"""Recursive function to construct binary of size len from
  Inorder traversal in[] and Preorder traversal pre[].  Initial values
  of inStrt and inEnd should be 0 and len -1.  The function doesn't
  do any error checking for cases where inorder and preorder
  do not form a tree """
def buildTree(inOrder, preOrder, inStrt, inEnd):

   if (inStrt > inEnd):
      return None

   # Pick current node from Preorder traversal using
   # preIndex and increment preIndex
   tNode = Node(preOrder[buildTree.preIndex])
   buildTree.preIndex += 1

   # If this node has no children then return
   if inStrt == inEnd :
      return tNode

   # Else find the index of this node in Inorder traversal
   inIndex = search(inOrder, inStrt, inEnd, tNode.data)

   # Using index in Inorder Traversal, construct left
```

```
    # and right subtrees
    tNode.left = buildTree(inOrder, preOrder, inStrt, inIndex-1)
    tNode.right = buildTree(inOrder, preOrder, inIndex + 1, inEnd)

    return tNode
```