private TreeNode buildTree(int[] input, int start, int end) {  
 /\* Return null when the indices cross, preventing duplicates/infinite loops \*/  
 if (start > end) {  
 return null;  
 }  
 /\* Find the middle index of the array/subarray being input \*/  
 int middleIndex = ((start + end) / 2);  
 /\* Create a new node with the value from the middle of the array/subarray \*/  
 TreeNode nd = new TreeNode(input[middleIndex]);  
 /\* Create a new subtree to the left using the lower half of the array \*/  
 nd.left = buildTree(input, start, middleIndex - 1);  
 /\* Create a new subtree to the right using the upper half of the array \*/  
 nd.right = buildTree(input, middleIndex + 1, end);  
 /\* Return the root of the entire tree \*/  
 return nd;  
}



