Problem Domain:

Given 2 binary trees, find the common values between them.

Algorithm:

- Traverse one of the trees
- Put each node in the dictionary
- traverse the second tree and call .ContainsKey on each node to see if it exists in the dictionary
- if ContainsKey, then push it to an array
- return the array.

Edge Cases:

- if there aren't any matches, what do we return?
- if a tree is empty
- if both trees are empty
- if all the nodes in a tree are same
- if all the nodes in both trees match
- if there is only one node in a tree

Big O

- Time: O(n)
- Space: O(n)

Pseudo Code:

```
MatchingValues(tree1, tree2)

<Dictionary> storedValues = new Dictionary<int, string>
int[] outputArr = [ ]
while tree1 != null
tree1 root .Add
node.left .Add
node.right .Add

while tree2 != null
tree2 root .ContainsKey ?
node.left .ContainsKey ?
if .ContainsKey push to outputArr

return outputArr
```

Visual