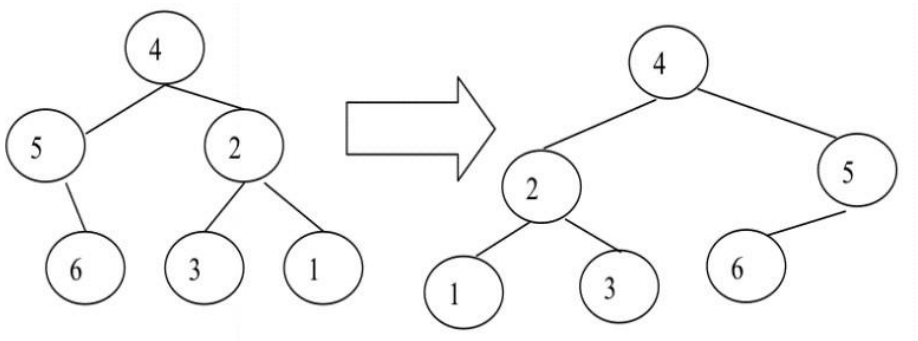
Binary Trees

---

1. Write a recursive function `mirrorTree()` that will modify a binary tree so that the resulting tree is a mirror image of the original structure.



   You should not create any intermediate or temporary trees. The function accepts a single parameter: a reference to the root node of the binary tree to be mirrored.

```
def mirrorTree(node):
```

2. Write a Python function `printSmallerValues()` that accepts a reference to the root node of a binary tree and prints all integers stored in the tree that are smaller than a given value `m`. The function definition is given as follows:

```
def printSmallerValues(node, m):
```

3. Write a function `smallest_value()` that returns the smallest value stored in a given tree. The function accepts a reference to the root of the given tree. You should determine the correct function definition.

4. Write a recursive function `hasGreatGrandchild()` that prints the values stored in all nodes of a binary tree that have at least one great-grandchild. The function accepts a single parameter: a reference to the root node of the binary tree.

```
def hasGreatGrandchild(node):
```

   Hint: Determine the common property shared by nodes with great-grandchild nodes and write a recursive function that computes that property.