CSC148 - Practice writing non-mutating Tree methods

We are going to write two methods, `leaves` and `average`. You'll find their APIs below, along with the key pieces of the `Tree` class that you need.

```
class Tree:
    """A recursive tree data structure.
    """
    # === Private Attributes ===
    # The item stored at this tree's root, or None if the tree is empty.
    _root: Optional[Any]
    # The list of all subtrees of this tree.
    _subtrees: List[Tree]

    # === Representation Invariants ===
    # - If self._root is None then self._subtrees is an empty list.
    #   This setting of attributes represents an empty tree.
    #
    #   Note: self._subtrees may be empty when self._root is not None.
    #   This setting of attributes represents a tree consisting of just one
    #   node.

    def __init__(self, root: Optional[Any], subtrees: List[Tree]) -> None:
        """Initialize a new Tree with the given root value and subtrees.

        If <root> is None, the tree is empty.
        Precondition: if <root> is None, then <subtrees> is empty.
        """

    def is_empty(self) -> bool:
        """Return whether this tree is empty.
        """

    def leaves(self) -> List:
        """Return a list of all of the leaf items in the tree.
        """

    def average(self) -> float:
        """Return the average of all the values in this tree.

        Return 0.0 if this tree is empty.

        Precondition: this is a tree of numbers.
        """
```

---

1. Consider method `leaves`. In the space below, identify each case that may need to be handled separately. For each, (a) describe the case and draw a tree that is an instance of it, (b) show each tree that must be recursed on and what the recursive call will return, and (c) show what the return value should be for this case.

(a) Case: description and tree	(b) Recursive calls: tree and return value	(c) Return value

2. Are there any cases that can be collapsed and handled in the same way?
3. Now go ahead and write method `leaves`.

Follow the same process to write method `average`.