Write recursive leaf_count function

first...

Read over the **_init_** method for class **Tree**:

```
class Tree:
    """
    A bare-bones Tree ADT that identifies the root with the entire tree.
    """
    def __init__(self, value=None, children=None):
        """
        Create Tree self with content value and 0 or more children

        @param Tree self: this tree
        @param object value: value contained in this tree
        @param list[Tree|None] children: possibly-empty list of children
        @rtype: None
        """
        self.value = value
        # copy children if not None
        self.children = children.copy() if children else []
```

next...

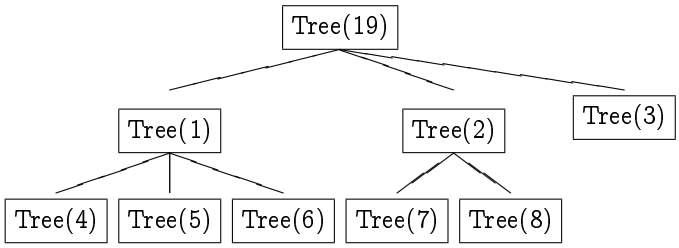
Now, read the header and docstring for the function **leaf_count**, and then answer the questions that follow it.

```
def leaf_count(t):
    """
    Return the number of leaves in Tree t.

    @param Tree t: tree to count the leaves of
    @rtype: int

    >>> t = Tree(7)
    >>> leaf_count(t)
    1
    >>> t = descendants_from_list(Tree(7), [0, 1, 3, 5, 7, 9, 11, 13], 3)
    >>> leaf_count(t)
    6
    """
```

then...

1. One of the examples in **leaf_count** docstring is simple enough not to require recursion. Write an **if...** expression that checks for this case, and then returns the correct thing. Include an **else...** for when the tree is *less* easy to deal with.

2. Below is a picture of a larger **Tree** with several levels. Consider a function call **leaf_count(t)**, supposing **t** refers to the root of the tree. Are there smaller trees for which it would be helpful to know their leaf count? Which smaller trees are they? Write an example of a function call **leaf_count(???)** on one of these smaller trees. Rather than **??**, You can access these trees through the variable **t**.



3. Suppose the call in the previous step gives you the correct answer according to the docstring: it returns the number of leaves for the tree. How will you combine the solutions for all the smaller instances to get a solution for **Tree t** itself? Write code to return the correct thing. Put this code in the **else...** expression that you created in the first step.