Create a generator that iterates through a tree by first yielding the root value and then yield each branch.

```
>> tree = Tree(1, [Tree(2, [Tree(5)]), Tree(3, [Tree(4)])])
>> print list(tree_sequence(tree))
[1, 2, 5, 3, 4]
```

The first part is easy. What do we want returned the first time? Just the root!

```
def tree_sequence(tree):
    yield tree.entry
```

Now we want to go through each branch, yielding the value at each node. Let's take it a step at a time. To iterate through branches we do:

```
def tree_sequence(tree):
    yield tree.entry
    for branch in tree.branches:
        Do something!
```

What do we want to do in each branch? Yield the value of each node! The function we are defining does that! So let's call tree_sequence on the branch. If our function works correctly (leap of faith) this should return a generator that yields the values of each branch. So we can iterate through the values yielded by tree_sequence(branch). We get:

```
def tree_sequence(tree):
    yield tree.entry
    for branch in tree.branches:
        for value in tree_sequence(branch):
```

Now what do we want to do for each value? Yield it!

```
def tree_sequence(tree):
    yield tree.entry
    for branch in tree.branches:
        for value in tree_sequence(branch):
            yield value
```

We're done!