I collaborated with:

Problem
Solve Problem 6 from Chapter 5 of your text. **Clarification**. What does *probe* mean? Imagine that for each vertex v in the tree, the label x_v is a quantity that must be computed and that is very time-consuming to compute x_v. A *probe* of v is a computation of x_v so we want to minimize the number of such operations.
Solution

Algorithm 1 Find local minimum

```
function MIN(T)
    if T is a leaf then
        return T.value
    else if T.value is less than both of its children then
        return T.value
    else
        min(smallest child)
    end if
end function
```

Proof. If T is a leaf then we know no edge is incident of it, so it is a local minimum. Moreover, if T has children, then we check if its smaller than them. We do not have to check its parent as our recursive call handles it. If T happens to be bigger than its children, then we know the children are smaller and do not have to check for the parent again. We keep going down the tree until we reach a leaf or a node that is a local minimum. Note that we only move down the tree if there is a smaller node.
This takes $O(\log n)$ as the most traversing we do is all the way to a leaf and the depth of a tree is $\log n$ □