I collaborated with: W3023339

Problem
Give a linear time, that is $O(n)$-time, algorithm to find the diameter of a tree $T = (V, E)$. Prove that your algorithm is correct. Suggestion. Consider modifying recursive DFS so that it also computes, for each vertex v

(a) The diameter of the subtree of T rooted at v

(b) The longest path from v to a leaf in the subtree of T rooted at v

Then show how, knowing this information for all children of some vertex u, we can determine this information for u itself. (Why do you think we need to know both (a) and (b)?)
Solution

Algorithm 1 Diameter of tree T

```
function DIAMETER(T)
    if T is null or leaf then return (0,0)
    else
        let S be an empty set
        for all neighbors v of T do
            add Diameter(v) to S
        end for
        B ← largest b in S + 1
        A ← if sum of two largest b's > largest a then b's else largest a
        return (A, B)
    end if
end function
```

Base Case: if we have a tree of only one node, then its diameter is zero. This algorithm holds as this base case is addressed.
Inductive Hypothesis: Assume this algorithm works for a tree T of up to k nodes, s.t. $k \in \mathbb{Z}, k \geq 1$.
Inductive Step: Now, add a node to T s.t. there are $k+1$ nodes in T. We can see that we can make a subtree containing the node we added. This subtree has less than $k+1$ nodes. According to our inductive hypothesis, this algorithm can compute the diameter of this subtree and we now need to compute the diameter for the larger subtree. The algorithm picks the diameter to be the two longest paths or the largest diameter from the root of the neighbors. This information allows us to assess whether the longest path from v to a leaf in the subtree of T rooted at v will form a larger diameter than that of the subtree, so it works for this subtree. Consequently, any other subtrees will have a diameter computed by the algorithm.