Q1.1 Prove reflect (reflect t) = t

Base case is when t is simply Empty, t = Empty

reflect (reflect t)
= reflect (reflect Empty)
= reflect (Empty)
= Empty = t
↳ the output equals t = Empty.

Induction step, when t has children (or subtrees) left, right.
t = Node (x, left, right)

From base case, reflect (reflect i) = i
So, reflect (reflect left) = left
reflect (reflect right) = right
} are induction hypothesis (IH)

reflect (reflect t)
= reflect (reflect (Node (x, left, right)))
= reflect (Node (x, reflect right, reflect left))
= Node (x, reflect (reflect left), reflect (reflect right))
= Node (x, left, right) = t □

(IH applies here)

Q1.2

Q1.2 Prove for all m:a' tree, size m = size' m 0.

Base case : m = Empty.

size m
= Size Empty = 0

size' m 0
= size' Empty 0 = 0

} From base case, size m = size' m 0

Induction step.

Induction hypothesis (IH) are when m has subtrees left, right.
(m = Node(x, left, right)

From base case, the IH are size left = size' left 0
size right = size' right 0.

size' m 0
= size' Node(x, left, right) 0
= size' left (size' right (1+0))
= size' left (size' right 1)
= size' left ((size' right 0) + 1) (IH applies here)
= size' left ((size right) + 1)
= size' left 0 + (size right) + 1) (IH applies here)
= size left + size right + 1
= 1 + size left + size right
= size m

□

Lemma is for all m: 'a tree, size m + acc = size' m acc

base case is when m is Empty and acc starts at 0.

size m = 0 because it returns 0

size' m 0 = 0 because it return acc, which is 0

Step case

size' Node(x,left,right) 0

= size' left (size' right (1+0))

= size' left (size' right 1)

= size' left ((size right) +1) by Lemma

= size left + ((size right) +1) by Lemma

= 1 + size left + size right

= size m