Question 4

1)

Base Case

Assume, `(g . h) x = (g (h x))`.

Show hold for `(Leaf x)`,

```
fmap (g . h) (Leaf x)
= Leaf $ (g . h) x              [ apply fmap def.  ]
= Leaf $ (g (h x))              [ apply (.) def.   ]
= fmap g (Leaf $ h x)           [ unapply fmap g   ]
= fmap h (fmap g (Leaf $ h x))  [ unapply fmap h   ]
= fmap g . fmap h (Leaf $ x)    [ unapply (.) def. ]
```

I.H

Assume holds for `lt` and `rt` in `(Node lt x rt)`

For lt,

```
fmap (g . h) lt = fmap g . fmap h lt
```

and rt,

```
fmap (g . h) rt = fmap g . fmap h rt
```

I.S

Prove for `fmap (g . h) (Node lt x rt) = fmap g . fmap h (Node lt x rt)`,

```
fmap (g . h) (Node lt x rt)
= Node (fmap (g . h) lt) ((g . h) x) (fmap (g . h) rt)        [ apply fmap def.  ]
= Node (fmap g . fmap h lt) ((g . h) x) (fmap g . fmap h rt)  [ by I.H           ]
= Node (fmap g (fmap h lt)) (g (h x)) (fmap g (fmap h rt))    [ apply (.) def.   ]
= fmap g (Node (fmap h lt) (h x) (fmap h rt))                 [ unapply fmap g   ]
= fmap g (fmap h (Node lt x rt))                              [ unapply fmap h   ]
= fmap g . fmap h (Node lt x rt)                              [ unapply (.) def. ]
```