
CS 440 — BST Activity

Mattox Beckman

Code

Here is the code that was given in the video.

```
data BST a = Node a (BST a) (BST a)
            | Empty
            deriving Show

add elt Empty = Node elt Empty Empty
add elt orig@(Node k lc rc)
  | elt == k    = orig
  | elt < k     = Node k (add elt lc) rc
  | otherwise  = Node k lc (add elt rc)

addlist xx = aux xx Empty
  where aux [] t      = t
        aux (x:xs) t = aux xs (add x t)

find desire Empty = False
find desire (Node k lc rc)
  | desire == k = True
  | desire < k  = find desire lc
  | otherwise   = find desire rc

del victim Empty = Empty
del victim orig@(Node k lc rc)
  | victim == k =
    case orig of
      Node _ Empty Empty -> Empty
      Node _ lc Empty    -> lc
      -- More code here!!

  | victim < k  = Node k (del victim lc) rc
  | otherwise  = Node k lc (del victim rc)
```

Questions

1. Finish the `del` function. You will need to write a helper function `getPred` to get the predecessor. Show your code to a classmate and convince them that it works.
2. If you finish quickly, see if you can design an AVL tree using this as a starting point.