Programming HW 9

CS 169 Spring 20, Linnell

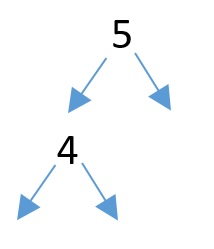
Scala Pattern Matching and Inheritance

Due Wed. 6/3 in class

Submission instructions at the end

1. (15 points) Add the following functions to the BSTree class we discussed Friday 5/29

def depth:Int  This function should return the number of values traversed between the root and the deepest leaf (inclusive).   That is, the tree below has depth 2.  You may NOT use pattern matching (the match/case syntax).



def exactsubtree(that:BSTree):Boolean   This function should return true if this appears in the exact same structure as a subtree of that (down to the Leaf references).  You MUST use pattern matching.

2.(5 points)  Write a function def findlast(xs:List[Int], x:Int):Int that returns the index of the last time the element x appears in the list xs.  Return -1 if the element does not appear in the list.  For this function, you MUST use pattern matching, and you may NOT use any built-in list functions.  I will give 2 points extra credit if you do it without a helper function.

**Submission instructions:** Submission instructions: Upload one file for each problem, named Problem1.txt, Problem2.txt.  To do this, make a copy of the .scala file, and rename it.