

Week 11 Exercises Part 1

Sandra Batista

1.1–1.2

Exercise 1: Creating a Balanced BST

1. Give an insertion order of the numbers 1, 2, 3, 4, 5, 6, 7 Such that the resulting BST is balanced and draw the BST.
 - Recall that a Balanced D-ary Tree: Tree where subtrees of any node differ in height by at most 1
 - $|\text{Height}(\text{right subtree of root}) - \text{Height}(\text{left subtree of root})| \leq 1$

Exercise 2: BST Find

Starter code:

https://github.com/sandraleeusc/csci104_fall2020_lecture/blob/master/bst_exercises.cpp

1. Write a recursive function to **find** a value in a BST.
2. Trace the function on the BST in main for a value not in the tree such as 5 and a value in the tree such as 3

Exercise 3: BST FindAllInRange

Starter code:

https://github.com/sandraleeusc/csci104_fall2020_lecture/blob/master/bst_exercises.cpp

1. Write a function, **findAllInRange**, that given a root to a balanced binary search tree and two values, k_1 and k_2 , prints out all the keys in the tree that are greater than or equal to k_1 and less than or equal to k_2 from smallest to largest. Your implementation must be recursive and run in $O(\log(n) + v)$ where v is the number of key values in the range and n is the total number of keys in the tree. You may use helper functions. No loops allowed.
2. Modify the code in main to test your function. You can also trace how your code should behave on your tests.