**Name:**  Ananya Singh

**Section:** 004

**Github URL:** github.com/ananyasingh7/Trees

**Name of ALL collaborators:** Miguel Meireles

**URLS I consulted:** <https://www.geeksforgeeks.org/>, <https://www.youtube.com/user/purpongie>

1 a) Any node in the BST has at most two children, a left child and right child

If the node has a right subtree, every value in the subtree is greater than the node’s value

If the node has a left subtree, every value in the subtree is less than the node’s value

b) **A close up of text on a white background

Description automatically generated**

c)

**findPrevRec**

1. Just to clarify, you want me to create a binary to tree where all the functions will be recursive?
2. One edge case that immediately pops up in my head is what happens if there isn’t a number that is smaller than the given node? Am I to prevent that from happening or will I simply return -1?
3. Check picture below
4. Check picture below

**A close up of text on a white background

Description automatically generated**

1. My algorithm uses inorder traversal which is recursive and requires implicit space.
2. See code on github
3. As much as it is easy to implement a recursive algorithm, I don’t think its efficient because of all the implicit space that is being used every time the function is called

2) a) [0005, 0006, 0007, 0010, 0011, 0012, 0016, 0017, 0018, 0019, 0020]

b)

A close up of text on a whiteboard

Description automatically generated

c) look at the trees both iterative and recursive, they both have it

3 a) look at github in the file named arrayOfIntegers.java

b) look at github in the file named arrayOfIntegers.java

4) a) Properties of BBST: Node has to have a balance factor of at most 1, every time you either insert or delete… the tree has to self-balance, the root has to be greater than both children and again like BST…you can have up to at most 2 children

b) h: height of tree

|  |  |  |
| --- | --- | --- |
| Insert | O(log h) |  |
| Delete | O(log h) |  |
| Find-next | O(log h) |  |
| Find-prev | O(log h) |  |
| Find-min | O(log h) |  |
| Find-max | O(log h) |  |
|  |  |  |
|  |  |  |

c) look at github in the file called AVL.java

5) a) look at github in the file called questionFive.java

b) Yes I got stack overflow because the recursive stack got too large while trying to insert and there is too much implicit space as well.

10 integers output:

A screenshot of a cell phone

Description automatically generated

c) look at the file called questionFive.java

6) a) look at github in the file called question6.java

b)

A screenshot of a cell phone

Description automatically generated

c)

A screenshot of a cell phone

Description automatically generated