Name: Inho Choi **ID:** 1801187.

Problem 1

Suppose you have a Binary Search Tree with N nodes whose preorder traversal is the same as its inorder traversal. What is the cost for finding the Min value from such a tree? Max value?

Such a BST will be linear and to the right. $O(1)$ to find the min, $O(N)$ to find the max If N is the number of nodes in the BST.

Problem 2

You want to maintain peoples' names and for each person, keep track of all the streets they lived on. Assume there are P total people and each person has lived on average E former streets. Assume you're using a map of name/streets : map< names, set< streets >> names2streets;

What is the Big-O cost of:

a. Finding the names of all people who lived on "Levering Street"?

$O(P\log_2 E)$.

b. Determining if "Bill" ever lived on "Westwood Blvd"?

$O(\log_2 P + \log_2 E)$

c. Printing out every name along with each person's street addresses in alphabetical order?

$O(PE)$

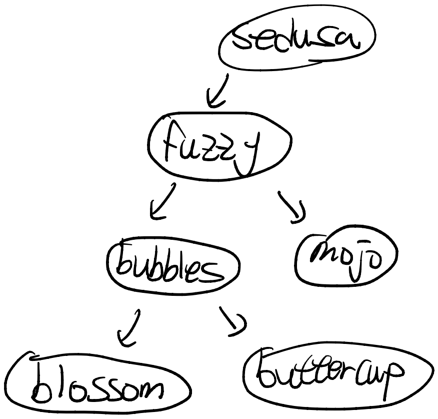
d. Printing out all the streets that "Tala" has lived on?

$O(\log_2 P + E)$

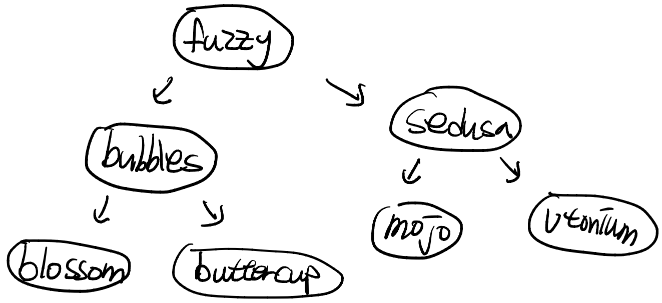
Name: Inho Choi **ID:** 180178n

Problem 3:

a. Draw a binary search tree that is created if the following strings are inserted into the tree in the given order: “sedusa”, “fuzzy”, “bubbles”, “utonium”, “buttercup”, “mojo”, “blossom”.



b. Draw a balanced binary search tree containing the same names given above (Just draw what the tree looks like, do not attempt a balancing algorithm).

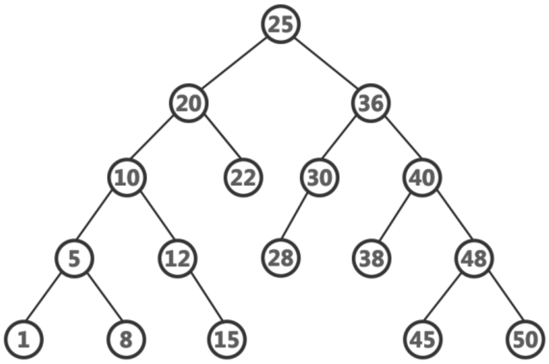


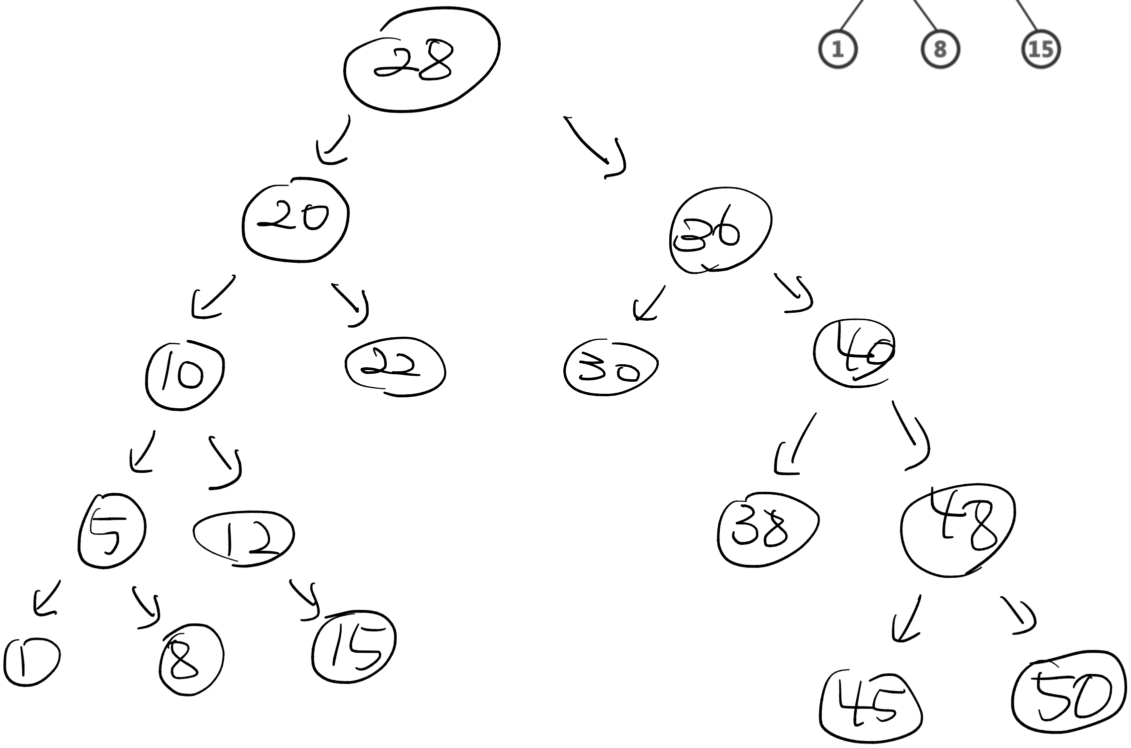
c. How would you describe the Big-O for searching a tree with a structure similar to the tree in part a? part b?

Part a for O(P)
Part b for O(logP)

Name: Inho Choi **ID:** 1801787.

Problem 6:
Consider the following Binary Search Tree, Show what the tree looks like after deleting the root node.


25
20
36
10
22
30
40
5
12
28
38
48
1
8
15
45
50


28
20
36
10
22
30
40
5
12
38
48
1
8
15
45
50