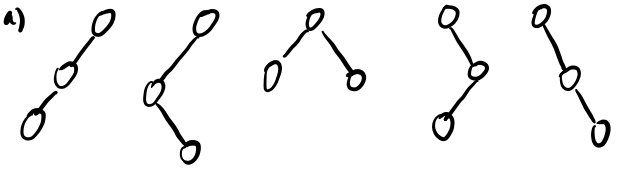
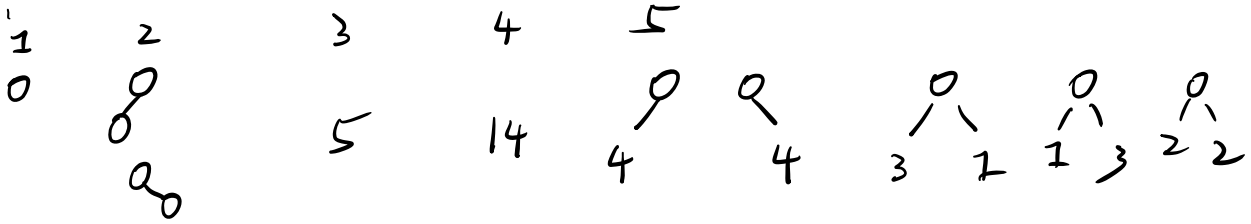
(a)



(b)



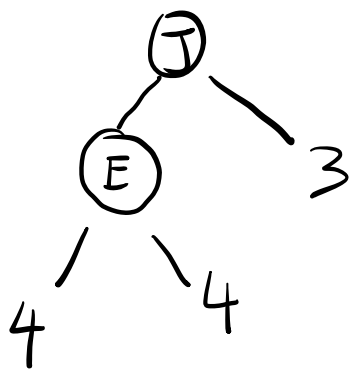
$n_1 = 1 \quad n_2 = 2 \quad n_3 = 5 \quad n_4 = 5 \times 2 + 2 \times 2$

$\Rightarrow n_5 = 14 \times 2 + 2 \times 5 + 2 \times 2$

$= 28 + 10 + 4 = 42$

2.

A B C D E F G H I J K L M



14 types of tree of 4

5 types of tree of 3

$14 \times 14 \times 5 = 980$

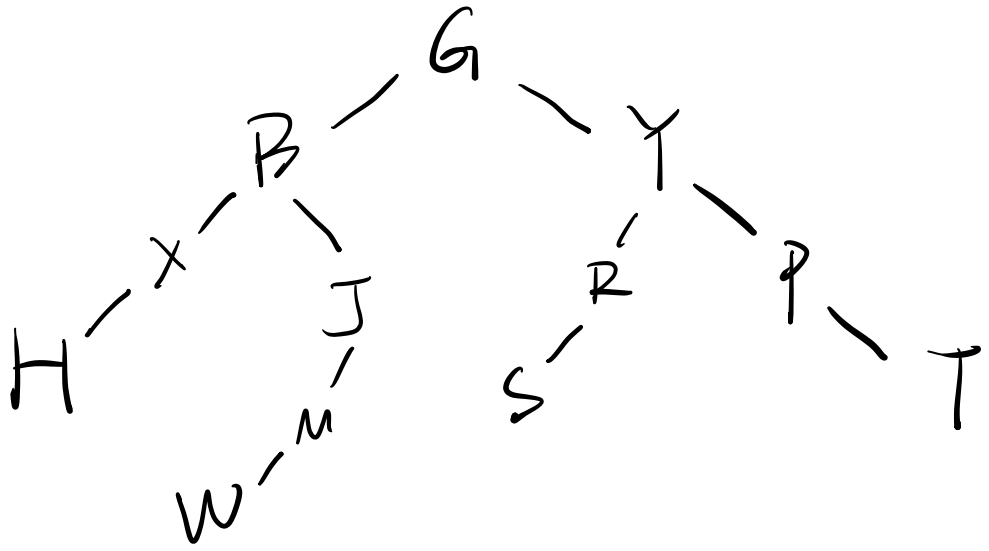
3. in order (left, Root, Right)

post order (left, Right, Root)

H X B W M J | G | S R Y P T

H X W M J B | S R T P Y G

L R



by pre order we know G is the root

Y is the Right child of root, and by both we know H is the most left child

and T is the Right most child.

4.

(a) (1) extraMin()

follow the left pointers from the root until a node with no left pointers. delete that node and return the value.

It has O(log n) time

(2) Insert()

Insert the key in the AVL Tree and rotate to rebalance. It also takes O(log n) time.

(2) It is worse than a heap

running time for min & extract min of heap is O(1)

insert has O(log n) which is the same for both,