Benjamin Swenson

1. **(15 points)** Consider the binary tree T shown in Fig. 1.

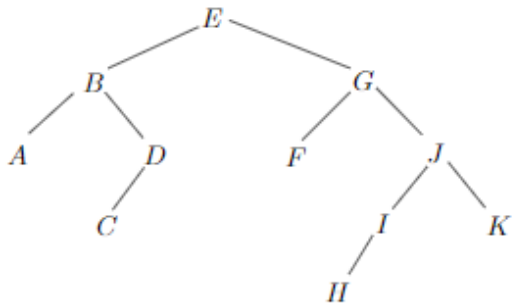


Figure 1: A binary tree T for Question 1

(a) Recall that the *depth* of a node v is defined to be the number of edges in the path from v to the root of T, and the *height* of a node v is defined to be the number of edges in the path from v to its deepest descendent leaf. The height of the tree is defined to be the height of the root.

What are the depth and the height of the node G? What is the height of the tree? **(6 points)**

Depth of G: 1
Height of G: 3
Height of Tree: 4

(b) Give the pre-order, in-order, and post-order traversal lists of T.

Pre-order: E, B, A, D, C, G, F, J, I, H, K
In-order: A, B, C, D, E, F, G, H, I, J, K
Post-order: A, C, D, B, F, H, I, K, J, G, E

2. **(10 points)** Consider the binary search tree in Fig. 2. Draw the new tree after remove operation: remove(47). As discussed in class, if v is the node you want to remove and v has two children, then you should use the smallest node in the **right** subtree of v to replace v.

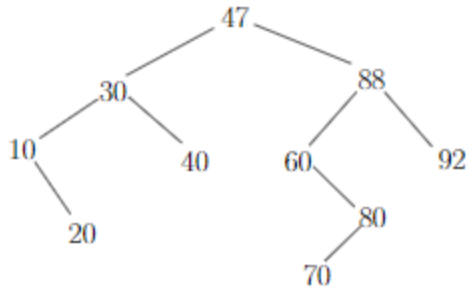
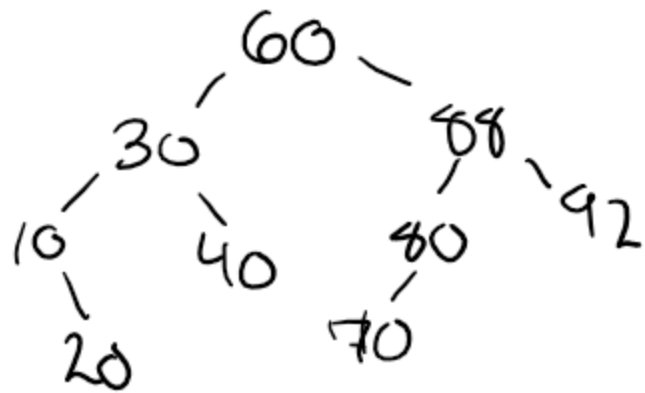


Figure 2: A binary search tree for Question 2

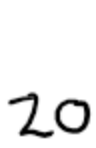
Remove (47)



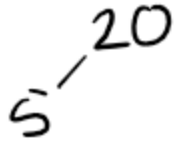
3. **(15 points)** Starting from an empty tree, draw the final binary search tree after the following sequence of operations: insert(20), insert(5), insert(15), insert(30), insert(40), insert(25), insert(10), insert(12), insert(8), insert(28), remove(20), insert(20), remove(5).

   Note: If you want to remove a node v that has two children, then, again, you are required to use the smallest node in the right subtree of v to replace v.

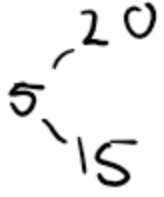
(1)



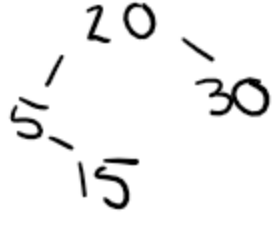
(2)



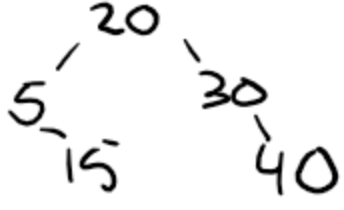
(3)



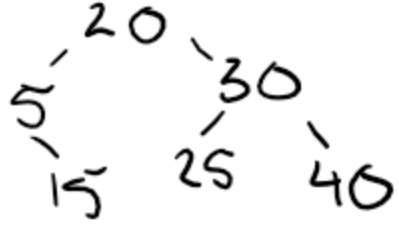
(4)



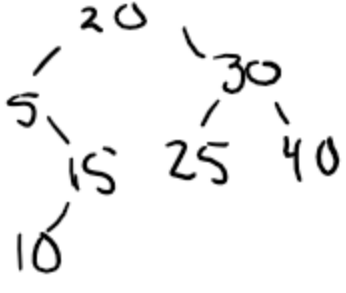
(5)



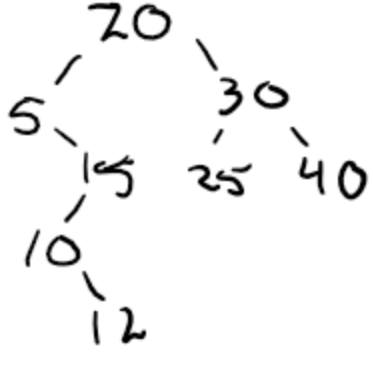
(6)



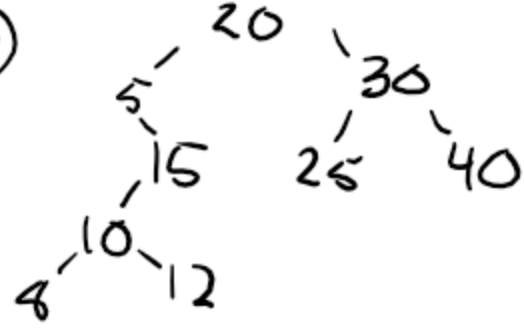
(7)



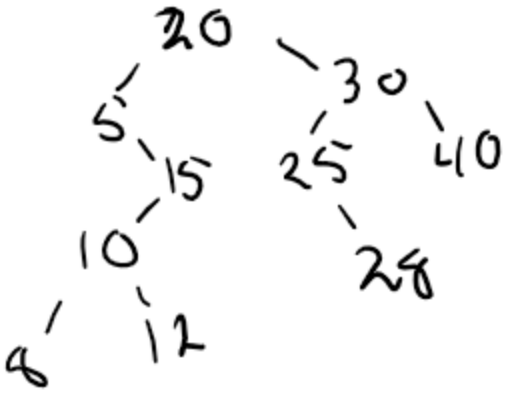
(8)



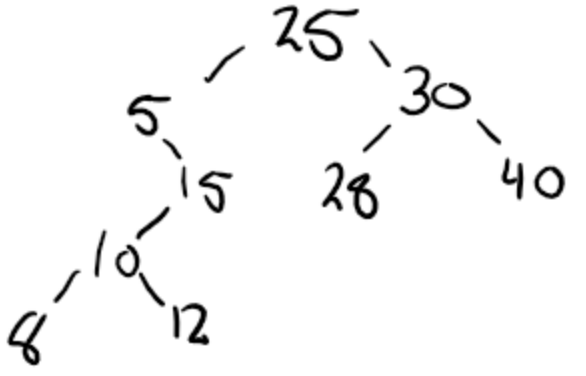
(9)



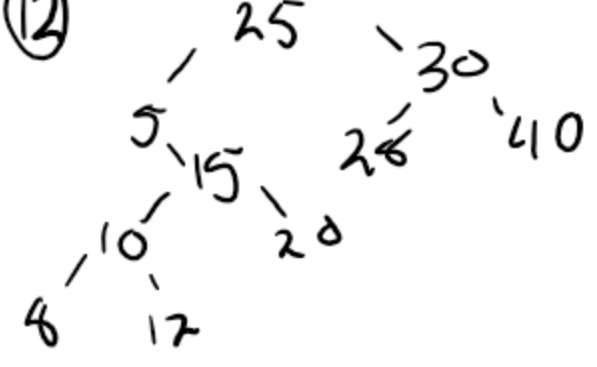
(10)


20
5
30
15
25
40
10
28
8
12

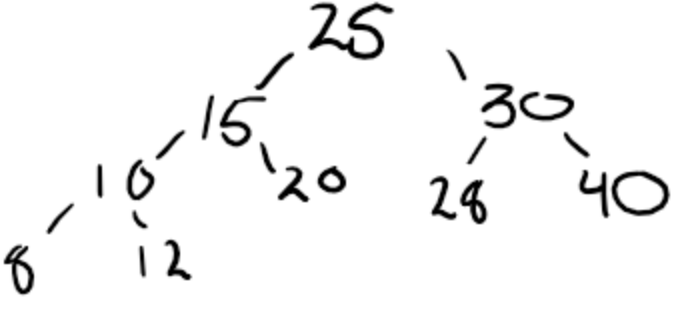
(11)


25
5
30
15
28
40
10
8
12

(12)


25
5
30
15
28
40
10
20
8
12

(13)


25
15
30
10
20
28
40
8
12