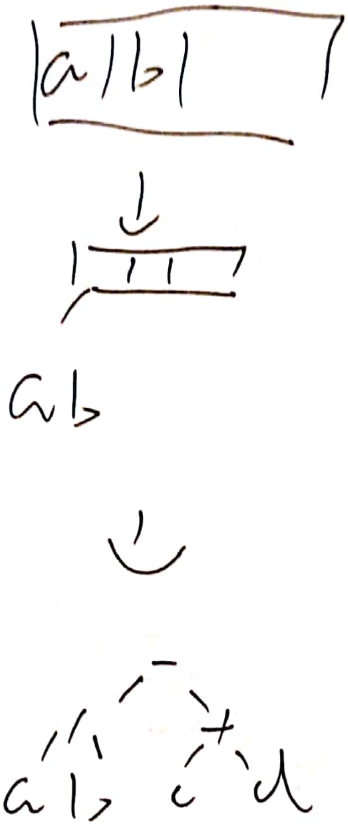
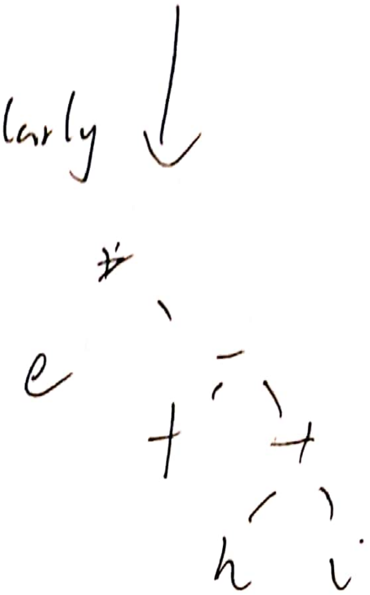
HW3 Yusi Xing 11596354

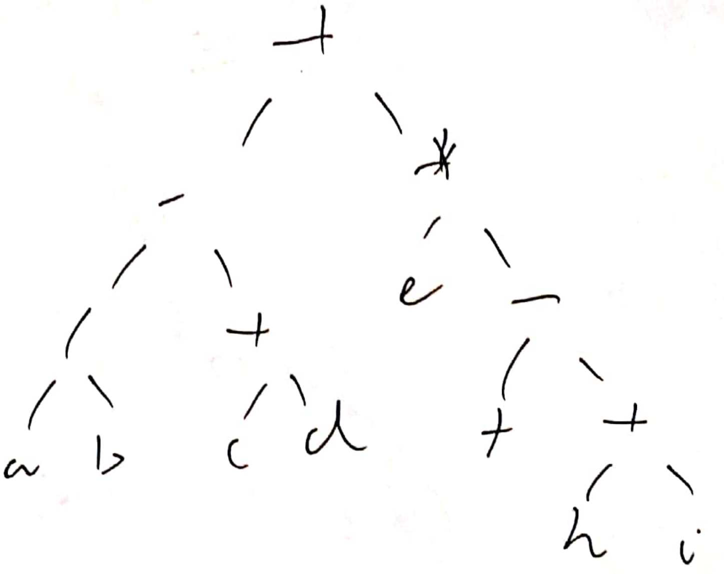
1. (a/b-c+d) + e*(f-h+i)



Similarly



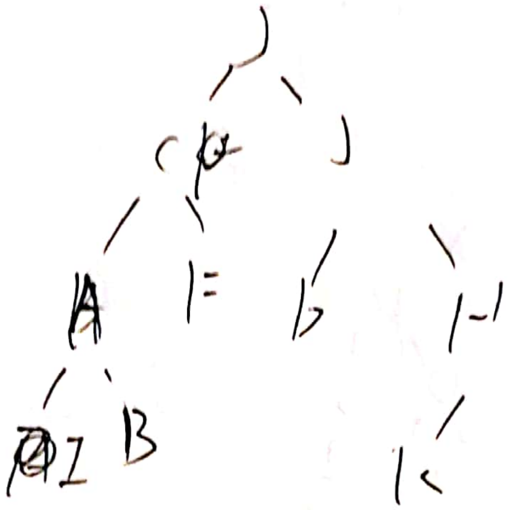
Combine, we get



2. I actually didn't figure it out in the exam..

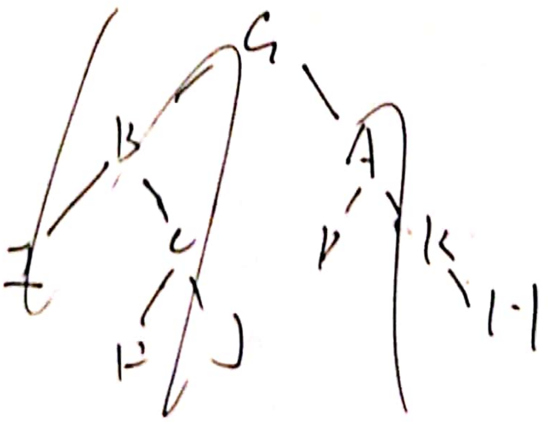
Sorry...

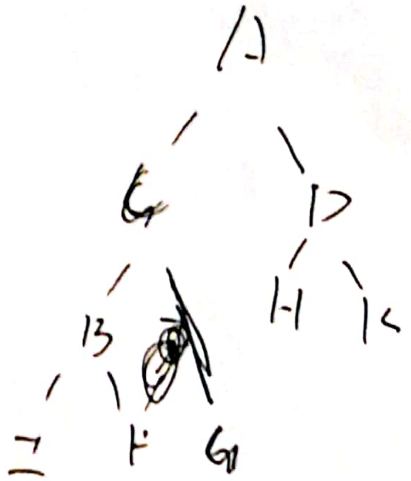
I think: based on the orders



I tried to find the root first, and I know I is the left leaf

wait..





I think I solved it!

3、 a. 4. Because from root to leaf.

b. 0, by definition.

c. ~~B~~. 0.

d. 3.

e. 0001. 0020. 0052. 0083. 0099.

0125. 0152.

f. 0.

5. (node left right) pre order:

~~0003, 0001, 0020,~~

0100, 0050, 0003, 0001, 0020, 0080, 0052, 0090, 0083, 0099,
0150, 0125, 0152.

post order:

0001, 0020, 0052, 0083, 0099, 0090, 0080, 0050,
0125, 0152, 0150, 0100.

in order:

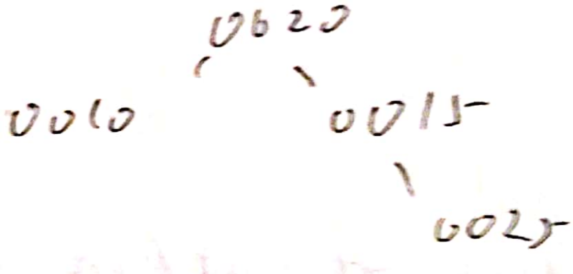
0001, 0003, 0020, 0050, ~~0080~~, 0052, 0099. 0090, 0080,
0100, 0125, 0150, 0152.

4. AVL tree is a type of binary search tree that for every node in it, the heights of its left and right differ by at most 1.

Because it is just like many other BST, easy searching and calculate, but it also reduce time complexity than a regular BST, it can also balance itself.

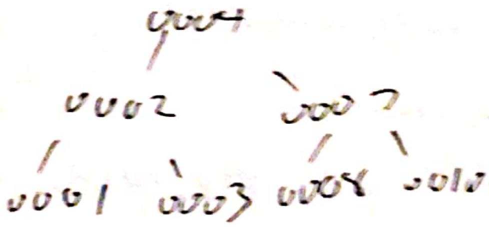
5. Based on deleting rules from presentation, ✓

we get:



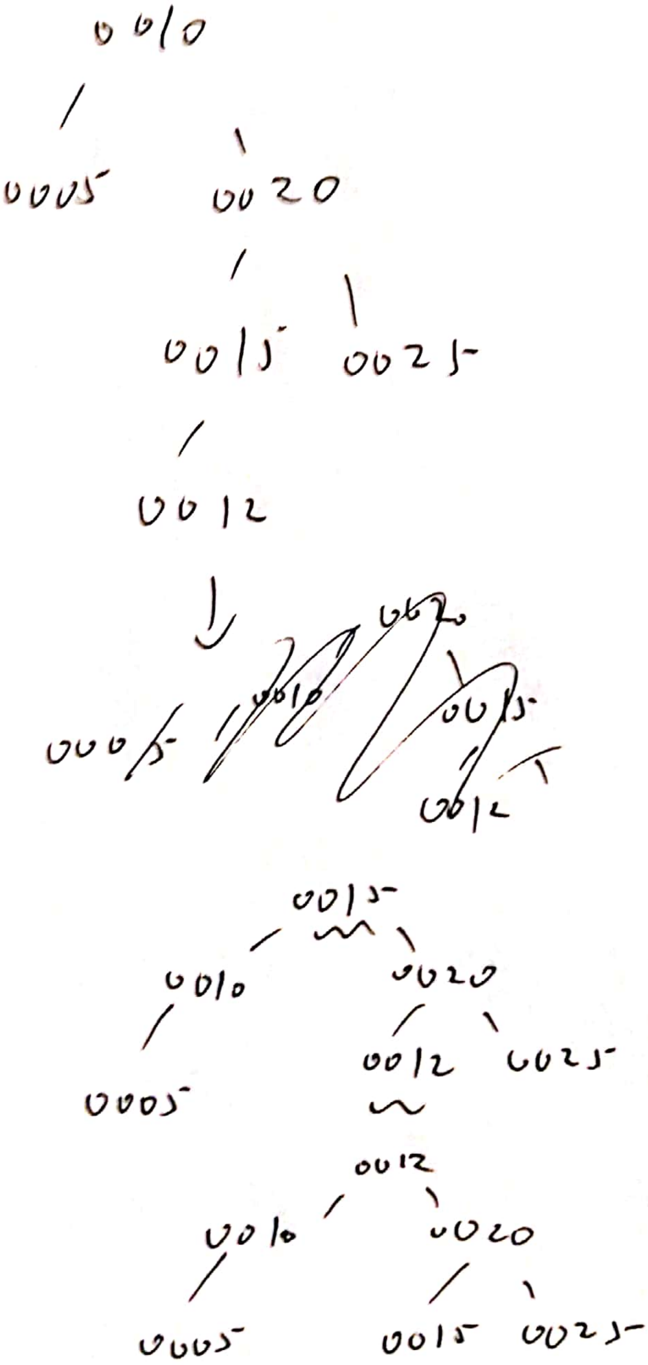
we rotate

6. Based on inserting rule from presentation. add in, then balance:



it is balanced.

7. Like question 6.



But we need to switch th
actually

balanced now.