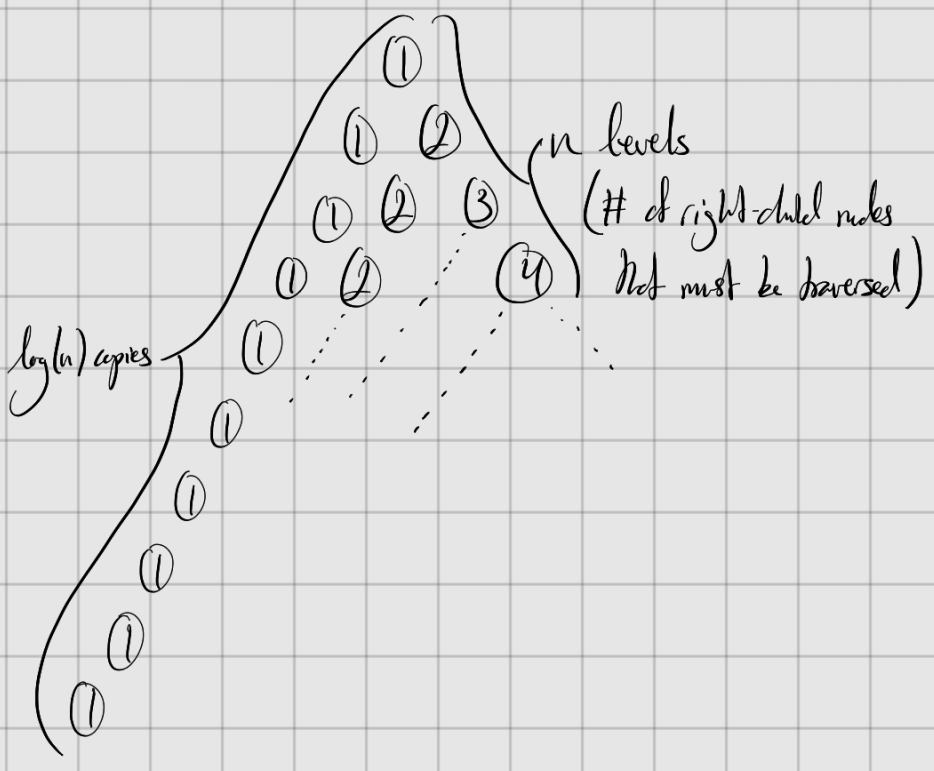
Jilin Zheng // U49758796

EC330 Homework 6

1. [Binary Search Trees, 10 points]
Specify and explain what is the worst-case asymptotic time needed to insert a sequence of log(n) copies each of the numbers 1 through n to an initially empty binary search tree. For example, if $n = 10^{10}$, then we will insert ten 1's, followed by ten 2's, followed by ten 3's, etc. Note that we are referring to a standard binary search tree, which is not necessarily balanced.

Worst-case runtime: $\Theta(n)$

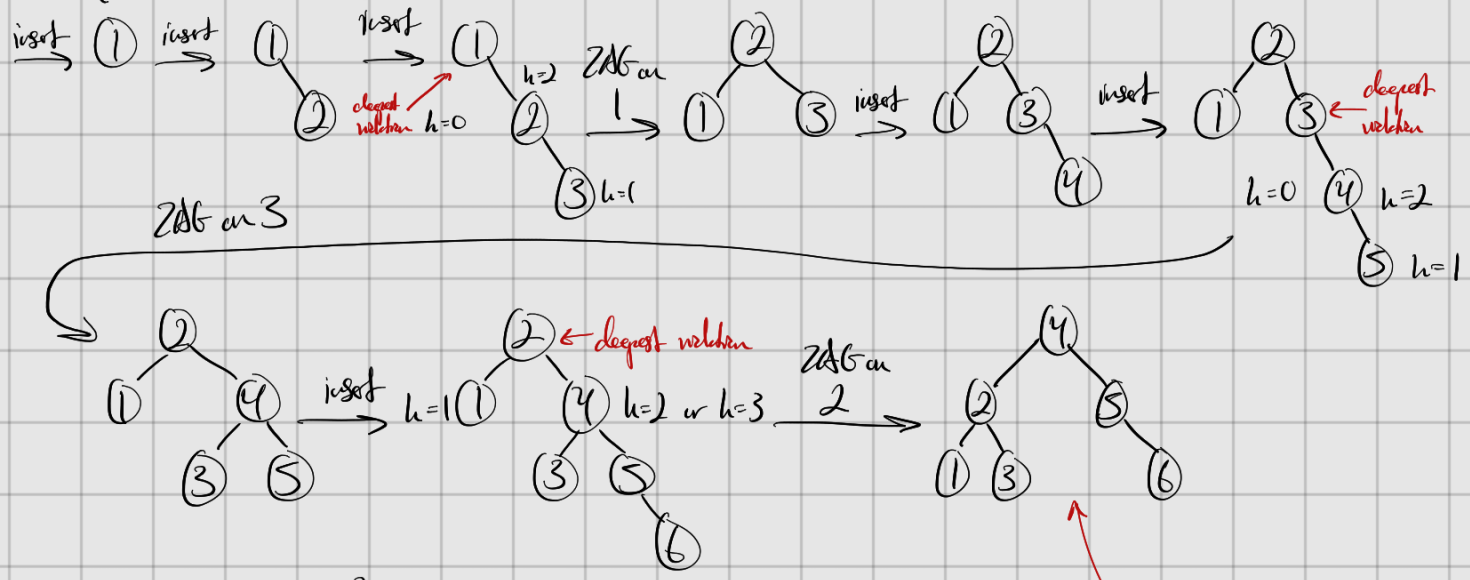
The height of the tree is $\log(n)+n$, where $\log(n)$ is the copies of the particular element being inserted (assuming that in the worst-case, each copy will be the left child of the previous copy), and n is the number of right-child nodes that must be traversed before beginning the insertion of a new element (and its copies). Asymptotically, $\log(n)+n=n$ since a linear function grows faster than a logarithmic function. Thus the worst-case asymptotic time is $\Theta(n)$.



2. [AVL Trees, 10 points]
Determine if the following claim is correct. It so, explain why, and if not, give a counter example: The order in which elements are inserted into an AVL tree does not matter, since the same AVL tree will be established following rotations.

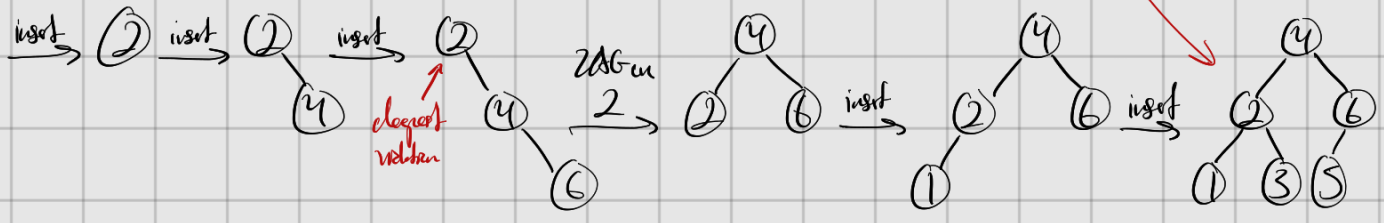
FALSE. Consider the following insertions into an initially empty AVL tree (left-to-right).

Insert {1, 2, 3, 4, 5, 6}



AVL on 3

Insert {2, 4, 6, 1, 3, 5}



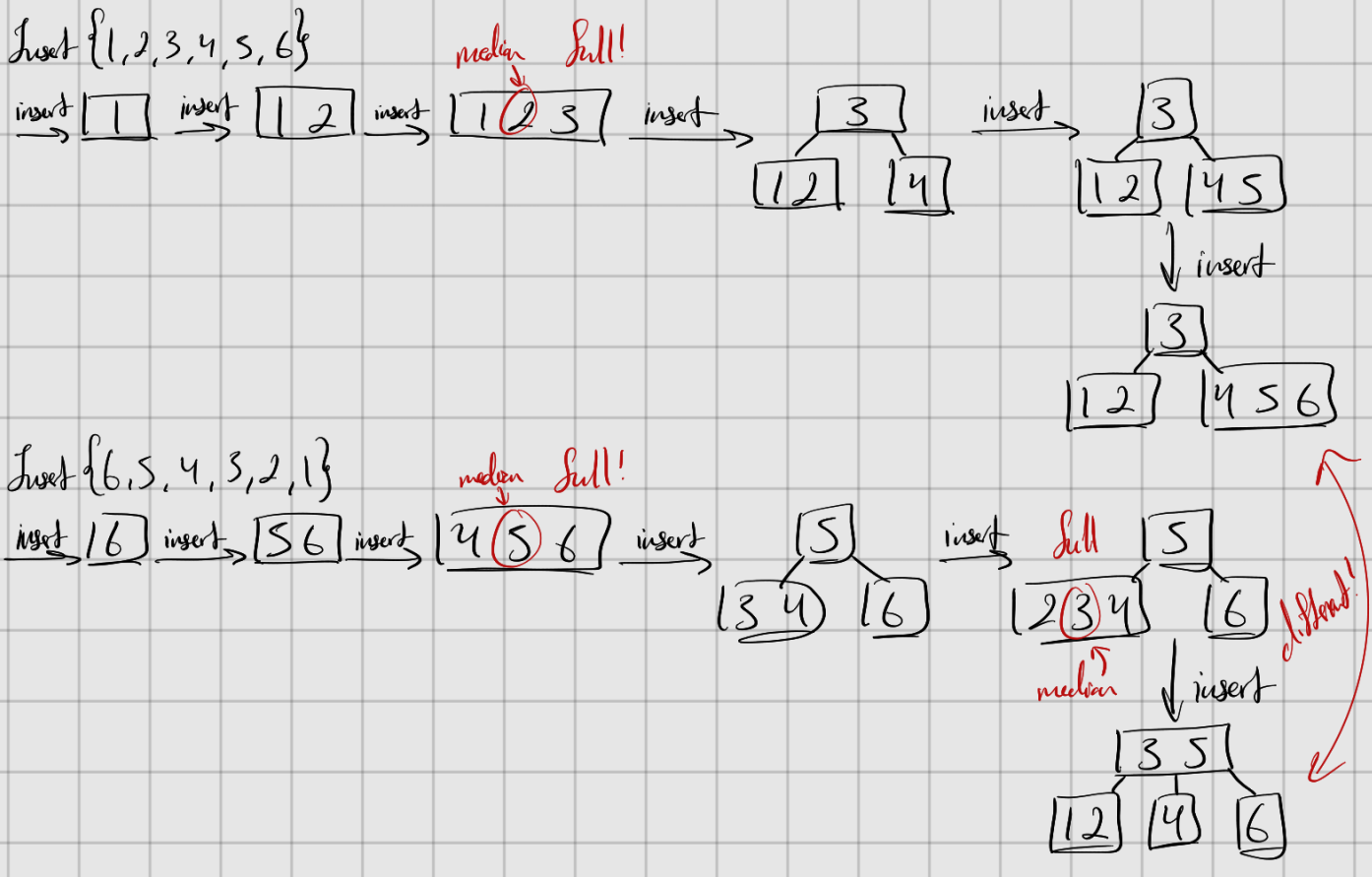
Different! Look @ 5 & 6

3. [B-tree, 10 points]
Determine if the following claim is correct. It so, explain why, and if not, give a counter example: The order in which elements are inserted into a B-tree does not matter, since the same B-tree will be established.

FALSE. Consider the following insertions into an initially empty B-tree (left-to-right).

Let degree = 2. Every node other than root has at least $2-1=1$ key and 2 children.

Every node has at most $2(2)-1=3$ keys and $2(2)=4$ children.

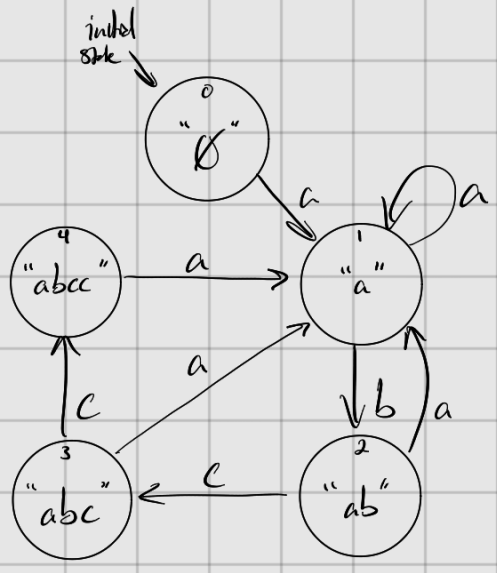


4. [String matching, 20 Points]

a. Draw the FSM to find the pattern "abcc". Assume that Σ={a,b,c}, and that any letter not in Σ returns to the initial state.

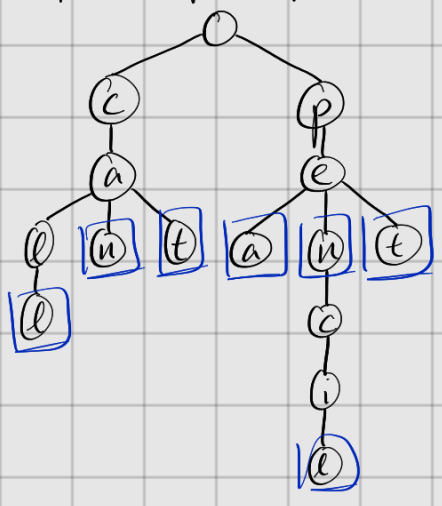
b. Draw a standard trie for the following set of strings: {pea, call, pencil, can, pet, cat, pen}.

a. "abcc", $\Sigma = \{a, b, c\}$



any input not in {a, b, c} OR not specified in the state's transitions goes to the initial state 0

b. {pea, call, pencil, can, pet, cat, pen}



- mark leaf node (a word exists)