

1) Suppose you are inserting a sequence of values into a binary search tree that is initially empty. You have no control over what order the elements arrive, so you must sometimes rebalance the tree according to the AVL technique as they are inserted. Consider the following two sequences, each starting with an empty tree:

- tree 1 is built from the sequence of values 8,7,5,4,3,6
- tree 2 is built from the sequence of values 7,8,9,10,11,12

For each element that is inserted, show

- the tree immediately after it is inserted, and before any rebalancing takes place
- Before each rotation, if any, you should indicate which node's AVL criterion is violated. Describe each rotation, if any, that is required. After each rotation, show the tree configuration.

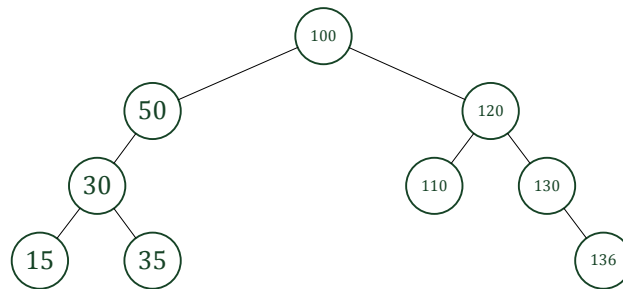
a) Tree 1 is built from the sequence of values 8, 7, 5, 4, 3, 6

Before	Inserting	After Insert	Violation Rule	Balanced Tree
	(8)	(8) <sup>EH</sup>	None	(8) <sup>EH</sup>
(8) <sup>EH</sup>	(7)		None	
	(5)		Left of Left	
	(4)		None	
	(3)		Right of Left	
	(6)		None	

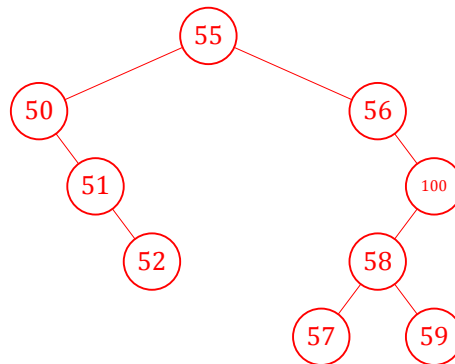
b) tree 2 is built from the sequence of values 7,8,9,10,11,12

Before	Inserting	After Insert	Violation Rule	Balanced Tree
	(7)	(7) <sup>EH</sup>	None	(7) <sup>EH</sup>
(7) <sup>EH</sup>	(8)	(7) <sup>LH</sup> └ (8) <sup>EH</sup>	None	(7) <sup>LH</sup> └ (8) <sup>EH</sup>
(7) <sup>LH</sup> └ (8) <sup>EH</sup>	(9)	(7) <sup>RRH</sup> └ (8) <sup>RH</sup> └ (9) <sup>EH</sup>	Right of Right	(8) <sup>EH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>EH</sup>
(8) <sup>EH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>EH</sup>	(10)	(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>RH</sup> └ (10) <sup>EH</sup>	None	(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>RH</sup> └ (10) <sup>EH</sup>
(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>RH</sup> └ (10) <sup>EH</sup>	(11)	(8) <sup>RRH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>RH</sup> └ (10) <sup>RH</sup> └ (11) <sup>EH</sup>	Right of Right	(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (10) <sup>EH</sup> ├ (9) <sup>EH</sup> └ (11) <sup>EH</sup>
(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (10) <sup>EH</sup> ├ (9) <sup>EH</sup> └ (11) <sup>EH</sup>	(12)	(8) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (10) <sup>RRH</sup> ├ (9) <sup>EH</sup> └ (11) <sup>RH</sup> └ (12) <sup>EH</sup>	Right of Right	(10) <sup>EH</sup> ├ (8) <sup>EH</sup> └ (11) <sup>RH</sup> ├ (7) <sup>EH</sup> └ (9) <sup>EH</sup> └ (12) <sup>EH</sup>

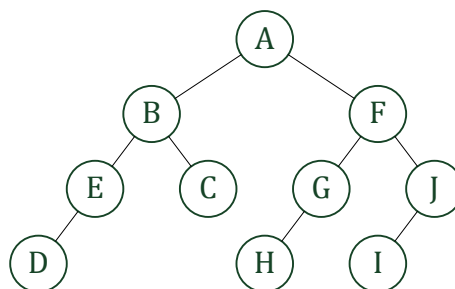
- 2) Suppose a breadth first search of a binary search tree gives the following sequence of values 100, 50, 120, 30, 110, 130, 15, 35, 136. Draw the binary search tree.



- 3) Suppose that a depth first search of a binary search tree gives the following sequence of values 55, 50, 51, 52, 56, 100, 58, 57, 59. Draw the binary search tree.



- 4) Suppose you are given the following binary tree.



What sequence of values would be obtained if the tree is traversed in

- inorder order?  
D, E, B, C, A, H, G, F, I, J
- postorder order?  
D, E, C, B, H, G, I, J, F, A
- preorder order?  
A, B, E, D, C, F, G, H, J, I
- BFS order?

A, B, F, E, C, G, J, D, H, I

e) DFS order?

A, B, E, D, C, F, G, H, J, I

5. (adapted from Goodrich and Tamassia, R-4.21)

For each function  $f(n)$  and for each time  $t$  in the following table, determine the largest size  $n$  of a problem  $P$  that can be solved in time  $t$  if the algorithm for solving  $P$  takes  $f(n)$  microseconds (one entry is already completed). Assume that a year has 365 days. Fill in the table. Explain your work for the 1 sec. column, including how the given entry is obtained. Caution: Take care with readability - BIG MESS with dull pencil == no marks!

$f(n)$	Time $t$			
	1 second	1 hour	30 days	100 years
$\log_2(n)$	$\sim 10^{300000}$	$\sim 10^{1000000000}$	$\sim 10^{800000000000}$	$\sim 10^{1,000,000,000,000,000}$
$n$	$10^6$	$\sim 3.6 \times 10^9$	$\sim 2.6 \times 10^{12}$	$\sim 3.12 \times 10^{15}$
$n \log_2(n)$	$\sim 10^5$	$\sim 10^9$	$\sim 10^{11}$	$10^{14}$
$n^2$	1000	60000	$\sim 1600000$	56000000
$2^n$	19	31	41	51

$$\begin{aligned}
 f(n) &= 1s \\
 f(n) &= 1,000,000\mu s \\
 \log_2(n) &= 1,000,000\mu s \\
 \frac{\log_{10}(n)}{\log_{10}(2)} &= 1,000,000\mu s \\
 \log_{10}(n) &= 1,000,000\mu s \times \log_{10}(2) \\
 n &= 10^{1,000,000\mu s \times \log_{10}(2)}
 \end{aligned}$$

$$\begin{aligned}
 f(n) &= 1s \\
 f(n) &= 1,000,000\mu s \\
 n &= 1,000,000\mu s
 \end{aligned}$$

$$\begin{aligned}
 f(n) &= 1s \\
 f(n) &= 1,000,000\mu s \\
 n^2 &= 1,000,000\mu s \\
 n &= \sqrt{1,000,000\mu s} \\
 n &= 1000
 \end{aligned}$$

$$\begin{aligned}
 f(n) &= 1s \\
 f(n) &= 1,000,000\mu s \\
 2^n &= 1,000,000\mu s \\
 n \log_{10}(2) &= \log_{10}(1,000,000\mu s) \\
 n &= \frac{\log_{10}(1,000,000\mu s)}{\log_{10}(2)} \\
 n &\approx 19.93
 \end{aligned}$$

$$\begin{aligned}
 f(n) &= 1s \\
 f(n) &= 1,000,000\mu s \\
 n \log_2(n) &= 1,000,000\mu s \\
 n \frac{\log_{10}(n)}{\log_{10}(2)} &= 1,000,000\mu s \\
 n \log_{10} n &= 1,000,000\mu s \times \log_{10}(2) \\
 n^n &= 10^{1,000,000\mu s \times \log_{10}(2)} \\
 n &= \sqrt[n]{10^{1,000,000\mu s \times \log_{10}(2)}}
 \end{aligned}$$

