



Practice questions on Height balanced/AVL Tree

Last Updated : 28 Dec, 2024

AVL tree is binary search tree with additional property that difference between height of left sub-tree and right sub-tree of any node can't be more than 1.

Here are some key points about [AVL trees](#):

- If there are n nodes in AVL tree, minimum height of AVL tree is $\text{floor}(\log_2 n)$.
- If there are n nodes in AVL tree, maximum height can't exceed $1.44 * \log_2 n$.
- If height of AVL tree is h , maximum number of nodes can be $2^{h+1} - 1$.
- Minimum number of nodes in a tree with height h can be represented as:
 $N(h) = N(h-1) + N(h-2) + 1$ for $n > 2$ where $N(0) = 1$ and $N(1) = 2$.
- The complexity of searching, inserting and deletion in AVL tree is $O(\log n)$.

We have discussed types of questions based on AVL trees.

Type 1: Relationship between number of nodes and height of AVL tree –

Given number of nodes, the question can be asked to find minimum and maximum height of AVL tree. Also, given the height, maximum or minimum number of nodes can be asked.

Que – 1. What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.

- (A) 2
- (B) 3
- (C) 4
- (D) 5

Solution: For finding maximum height, the nodes should be minimum at each level. Assuming height as 2, minimum number of nodes required: $N(h) = N(h-1)$

```

graph TD
    A(( )) --- B(( ))
    A --- C(( ))
    B --- D(( ))
    B --- E(( ))
    C --- F(( ))
    D --- G(( ))
  
```

- (A) $2 \cdot \log n$
- (B) $1.44 \cdot \log n$
- (C) Depends upon implementation
- (D) $\theta(n)$

```

graph TD
    A(( )) --- B(( ))
    A --- C(( ))
    B --- D(( ))
    B --- E(( ))
    C --- F(( ))
    D --- G(( ))
  
```

Checking for option (B). $1.44 * \log 7 = 4$, which is near to 3.

Out of these, option (B) is the best possible answer.

Type 2: Based on complexity of insertion, deletion and searching in AVL tree

–

Que – 3. Which of the following is TRUE?

- **(A)** The cost of searching an AVL tree is $\theta(\log n)$ but that of a binary search tree is $O(n)$
- **(B)** The cost of searching an AVL tree is $\theta(\log n)$ but that of a complete binary tree is $\theta(n \log n)$
- **(C)** The cost of searching a binary search tree is $O(\log n)$ but that of an AVL tree is $\theta(n)$
- **(D)** The cost of searching an AVL tree is $\theta(n \log n)$ but that of a binary search tree is $O(n)$

Solution: AVL tree's time complexity of searching, insertion and deletion = $O(\log n)$. But a binary search tree, may be skewed tree, so in worst case BST searching, insertion and deletion complexity = $O(n)$.

Que – 4. The worst case running time to search for an element in a balanced in a binary search tree with $n \cdot 2^n$ elements is

- (A) $\Theta(n \log n)$
- (B) $\Theta(n 2^n)$
- (C) $\Theta(n)$
- (D) $\Theta(\log n)$

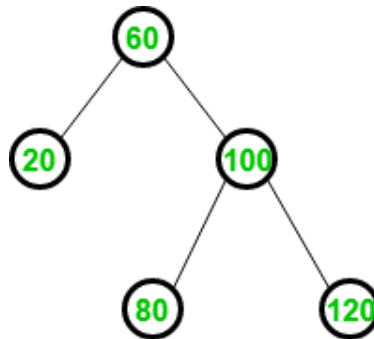
Solution: Time taken to search an element is $\Theta(\log n)$ where n is number of elements in AVL tree. As number of elements given is $n \cdot 2^n$, the searching complexity will be $\Theta(\log(n \cdot 2^n))$ which can be written as:

$$\begin{aligned}
 &= \theta(\log(n \cdot 2^n)) \\
 &= \theta(\log(n)) + \theta(\log(2^n)) \\
 &= \theta(\log(n)) + \theta(n \log(2)) \\
 &= \theta(\log(n)) + \theta(n)
 \end{aligned}$$

As $\log n$ is asymptotically smaller than n , $\theta(\log(n)) + \theta(n)$ can be written as

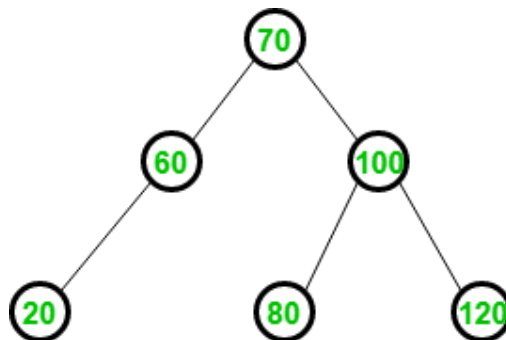
Type 3: Insertion and Deletion in AVL tree – The question can be asked on the resultant tree when keys are inserted or deleted from AVL tree. Appropriate rotations need to be made if balance factor is disturbed.

Que – 5. Consider the following AVL tree.

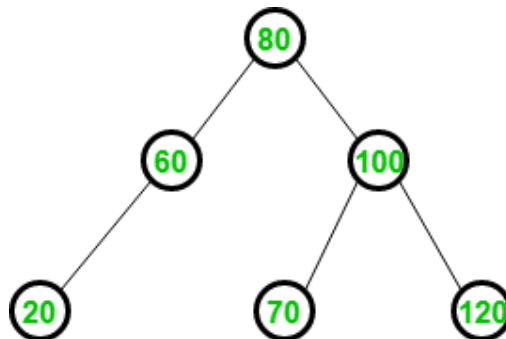


Which of the following is updated AVL tree after insertion of 70?

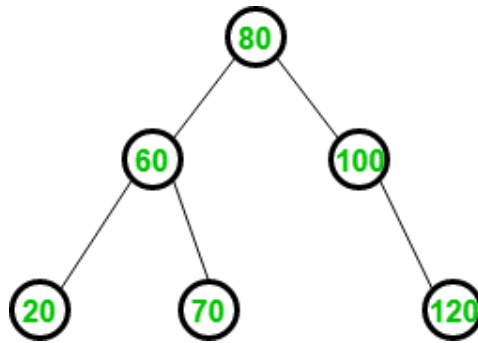
- (A)



- (B)

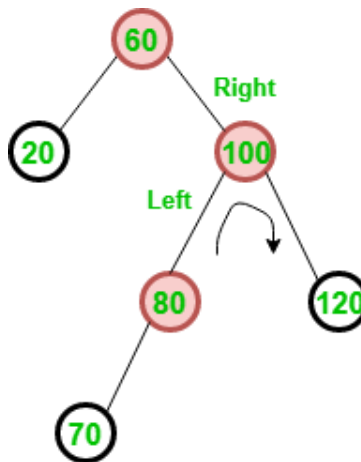


- (C)

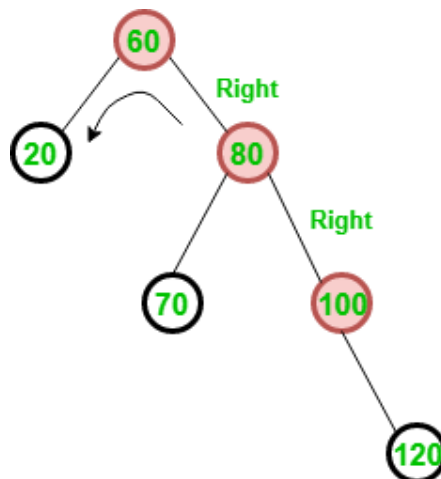


- (D) None

Solution: The element is first inserted in the same way as BST. Therefore after insertion of 70, BST can be shown as:



However, balance factor is disturbed requiring RL rotation. To remove RL rotation, it is first converted into RR rotation as:



After removal of RR rotation, AVL tree generated is same as option (C).