

What is an AVL Tree?

This lesson is a brief introduction to AVL trees, why they are used, and what makes them more efficient than regular binary search trees.

We'll cover the following ^

- Introduction
- Time Complexity

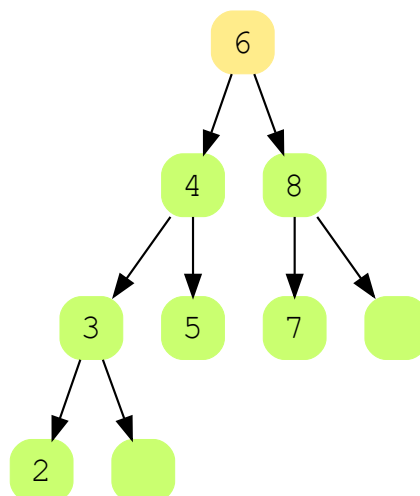
Introduction

Named after inventors **Adelson-Velsky** and **Landi** in 1962, they claimed that AVL trees are “An algorithm for the organization of information.” They are Binary Search Trees such that for every internal node v of the tree T , the heights of v 's children can differ by at most 1. To put it simply, for each Node, the height of the left and right subtrees in an AVL tree can differ at most by one or the tree is *balanced*. If at any point their difference becomes more than one, steps are taken to re-balance the tree.

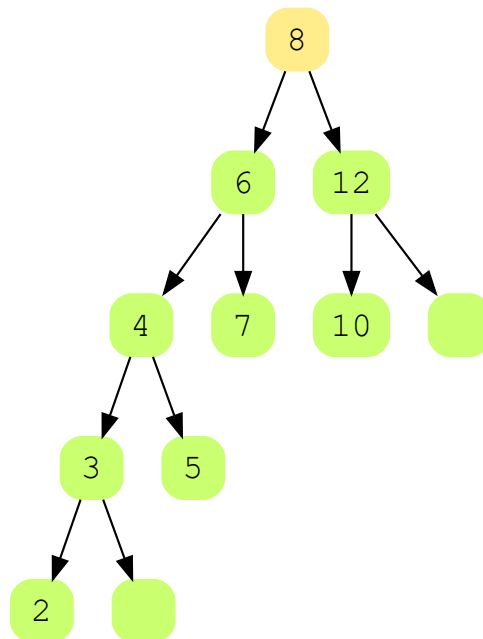
Time Complexity

In case of binary search trees, the time complexity of all three basic operations, Insertion, Deletion, and Search, take $O(h)$ time, where “ h ” is the height of the Binary Search Tree. The worst case time complexity is $O(n)$, for skewed BSTs, where “ n ” is the number of nodes in the tree. This is the same as an array or list. However, in the best case, when the tree is completely balanced, the time complexity for basic operations is $O(\log(n))$. AVL trees are essentially that: BSTs in the best-case.

The following diagram is an example of a valid AVL Tree



The tree in above figure is an AVL Tree as it follows the property of each node having a difference of height of their subtrees at most one. Given below is an example of an invalid tree as the height between siblings nodes exceed one.



The next lesson will cover basic operations performed in AVL trees. As the operations involve a series of steps and cases to follow to fulfill the property of AVL trees and keep the tree balanced, we will look into each of these operations individually.

[← Back](#)[In-Order Traversal](#)[Next →](#)[AVL Insertion](#)☒ **Mark as Completed**[Report an Issue](#)[Ask a Question](#)https://discuss.educative.io/tag/what-is-an-avl-tree__introduction-to-trees__data-structures-for-coding-interviews-in-python