

Self-Balancing-Binary-Search-Trees (Comparisons)

A computer science portal for geeks

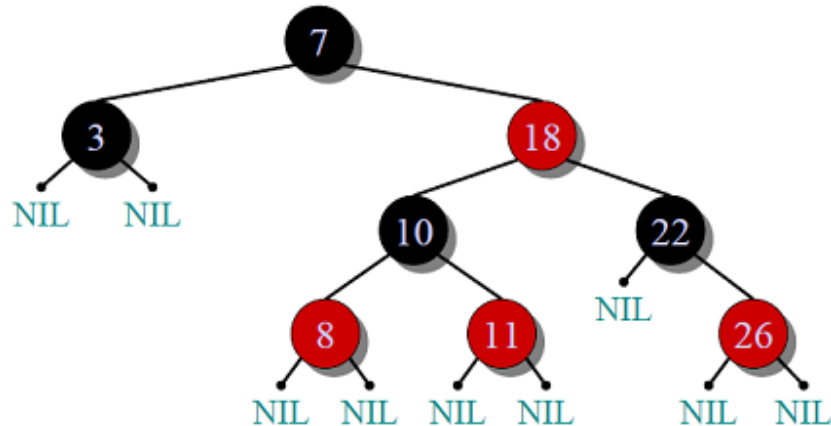
Self-Balancing Binary Search Trees are *height-balanced* binary trees. They automatically keep height as small as possible when insertion and deletion operations are performed on tree. The height is typically maintained in order of $\log n$ so that all operations take $O(\log n)$ time on average.

automatically keeps height as small as possible
The height is typically maintained in order of $\log n$ so that all

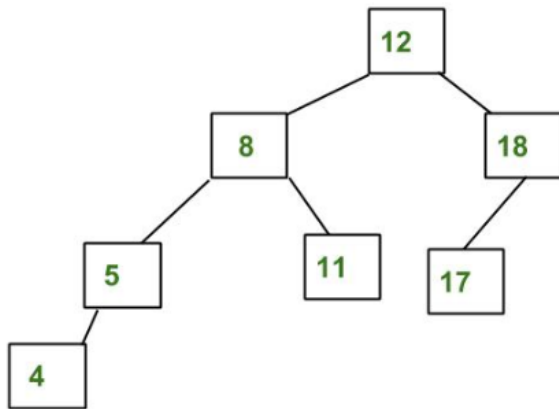
[Hire with us!](#)

Examples :

Red Black Tree



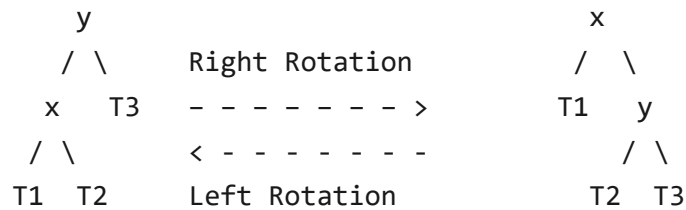
AVL Tree:



How do Self-Balancing-Tree maintain height?

A typical operation done by trees is rotation. Following are two basic operations that can be performed to re-balance a BST without violating the BST property ($\text{keys}(\text{left}) < \text{key}(\text{root}) < \text{keys}(\text{right})$). 1) Left Rotation 2) Right Rotation

T1, T2 and T3 are subtrees of the tree rooted with y (on the left side) or x (on the right side)



Keys in both of the above trees follow the following order

$\text{keys}(T1) < \text{key}(x) < \text{keys}(T2) < \text{key}(y) < \text{keys}(T3)$

So BST property is not violated anywhere.

We have already discussed **AVL tree**, **Red Black Tree** and **Splay Tree**. In this article, we will compare the efficiency of these trees:

METRIC	RB TREE	AVL TREE	SPLAY TREE
Insertion in worst case	$O(1)$	$O(\log n)$	Amortized $O(\log n)$
Maximum height of tree	$2 * \log(n)$	$1.44 * \log(n)$	$O(n)$
Search in worst case	$O(\log n)$, Moderate	$O(\log n)$, Faster	Amortized $O(\log n)$, Slower
Efficient Implementation requires	Three pointers with color bit per node	Two pointers with balance factor per node	Only two pointers with no extra information

METRIC	RB TREE	AVL TREE	SPLAY TREE
Deletion in worst case	$O(\log n)$	$O(\log n)$	Amortized $O(\log n)$
Mostly used	As universal data structure	When frequent lookups are required	When same element is retrieved again and again
Real world Application	Multiset, Multimap, Map, Set, etc.	Database Transactions	Cache implementation, Garbage collection Algorithms



Recommended Posts:

Comparisons involved in Modified Quicksort Using Merge Sort Tree

Second minimum element using minimum comparisons

Counting Inversions using Ordered Set and GNU C++ PBDS

Shortest Path Faster Algorithm

Total number of BSTs using array elements

Ackermann Function

Count the number of words with given prefix using Trie

Vertical and Horizontal retrieval (MRT) on Tapes

Duplicates Removal in Array using BST

Build a segment tree for N-ary rooted tree

Minimum Cost Graph

Count of different groups using Graph

Count number of increasing sub-sequences : $O(N \log N)$

Median of sliding window in an array



Abhishek rajput

Check out this Author's [contributed articles](#).

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Article Tags : [Advanced Data Structure](#) [Binary Search Tree](#) [AVL-Tree](#)

Practice Tags : [Binary Search Tree](#) [AVL-Tree](#)



1

3.6

☐ To-do ☐ Done

Based on 3 vote(s)

[Feedback/ Suggest Improvement](#)[Add Notes](#)[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

[Load Comments](#)

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

[About Us](#)
[Careers](#)
[Privacy Policy](#)
[Contact Us](#)

PRACTICE

[Courses](#)
[Company-wise](#)
[Topic-wise](#)
[How to begin?](#)

LEARN

[Algorithms](#)
[Data Structures](#)
[Languages](#)
[CS Subjects](#)
[Video Tutorials](#)

CONTRIBUTE

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)

@geeksforgeeks, Some rights reserved