



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages

[فارسی](#)

[Polski](#)

[Српски / srpski](#)

[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search

Weight-balanced tree

From Wikipedia, the free encyclopedia

For other uses, see [Optimal binary search tree](#).

In [computer science](#), **weight-balanced binary trees** (**WBTs**) are a type of [self-balancing binary search trees](#) that can be used to implement [dynamic sets](#), [dictionaries](#) (maps) and sequences.^[1] These trees were introduced by Nievergelt and Reingold in the 1970s as **trees of bounded balance**, or **BB[α] trees**.^{[2][3]} Their more common name is due to [Knuth](#).^[4]

Like other self-balancing trees, WBTs store bookkeeping information pertaining to balance in their nodes and perform [rotations](#) to restore balance when it is disturbed by insertion or deletion operations. Specifically, each node stores the size of the subtree rooted at the node, and the sizes of left and right subtrees are kept within some factor of each other. Unlike the balance information in [AVL trees](#) (which store the height of subtrees) and [red-black trees](#) (which store a fictional "color" bit), the bookkeeping information in a WBT is an actually useful property for applications: the number of elements in a tree is equal to the size of its root, and the size information is exactly the information needed to implement the operations of an [order statistic tree](#), viz., getting the n 'th largest element in a set or determining an element's index in sorted order.^[5]

Weight-balanced trees are popular in the [functional programming](#) community and are used to implement sets and maps in [MIT Scheme](#), [SLIB](#) and implementations of [Haskell](#).^{[6][4]}

Description [\[edit\]](#)

A weight-balanced tree is a binary search tree that stores the sizes of subtrees in the nodes. That is, a node has fields

- *key*, of any ordered type
- *value* (optional, only for mappings)
- *left*, *right*, pointer to node
- *size*, of type integer.

By definition, the size of a leaf (typically represented by a `nil` pointer) is zero. The size of an internal node is the sum of sizes of its two children, plus one ($\text{size}[n] = \text{size}[n.\text{left}] + \text{size}[n.\text{right}] + 1$). Based on the size, one defines the weight as either equal to the size, or as $\text{weight}[n] = \text{size}[n] + 1$.

Operations that modify the tree must make sure that the weight of the left and right subtrees of every node remain within some factor α of each other, using the same rebalancing operations used in AVL trees: rotations and double rotations. Formally, node balance is defined as follows:

A node is balanced if $\text{weight}[n.\text{left}] \geq \text{weight}[n]$ and $\text{weight}[n.\text{right}] \geq \text{weight}[n]$.^[7]

Lower values mean more balanced trees, but not all values of α are appropriate; Nievergelt and Reingold proved that

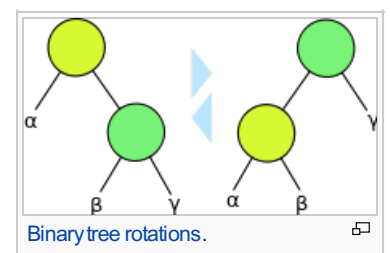
$$\alpha < 1 - \frac{1}{\sqrt{2}}$$

is a necessary condition for the balancing algorithm to work. Later work showed a lower bound of $2/11$ for α , although it can be made arbitrarily small if a custom (and more complicated) rebalancing algorithm is used.^[7]









Applying balancing correctly guarantees a tree of n elements will have height^[7]

$$h \leq \log_{\frac{1}{1-\alpha}} n = \frac{\log_2 n}{\log_2 \left(\frac{1}{1-\alpha} \right)} = O(\log n)$$

The number of balancing operations required in a sequence of n insertions and deletions is linear in n , i.e., constant in an [amortized](#) sense.^[8]



References [\[edit\]](#)

1. [^] Tsakalidis, A. K. (1984). "Maintaining order in a generalized linked list". *Acta Informatica* **21**: 101. doi:[10.1007/BF00289142](https://doi.org/10.1007/BF00289142) .
2. [^] Nievergelt, J.; Reingold, E. M. (1973). "Binary Search Trees of Bounded Balance". *SIAM Journal on Computing* **2**: 33. doi:[10.1137/0202005](https://doi.org/10.1137/0202005) .
3. [^] Black, Paul E. "BB(α) tree" . *Dictionary of Algorithms and Data Structures*. NIST.
4. [^] ^a ^b Hirai, Y.; Yamamoto, K. (2011). "Balancing weight-balanced trees". *Journal of Functional Programming* **21** (3): 287. doi:[10.1017/S0956796811000104](https://doi.org/10.1017/S0956796811000104) .
5. [^] Roura, Salvador (2001). *A new method for balancing binary search trees*. [ICALP](#). Lecture Notes in Computer Science **2076**. pp. 469–480. doi:[10.1007/3-540-48224-5_39](https://doi.org/10.1007/3-540-48224-5_39) . ISBN 978-3-540-42287-7.
6. [^] Adams, Stephen (1993). "Functional Pearls: Efficient sets—a balancing act". *Journal of Functional Programming* **3** (4): 553–561. doi:[10.1017/S095679680000885](https://doi.org/10.1017/S095679680000885) .
7. [^] ^a ^b ^c Brass, Peter (2008). *Advanced Data Structures*. Cambridge University Press. pp. 61–71.
8. [^] Blum, Norbert; Mehlhorn, Kurt (1980). "On the average number of rebalancing operations in weight-balanced trees"  (PDF). *Theoretical Computer Science* **11** (3): 303–320. doi:[10.1016/0304-3975\(80\)90018-3](https://doi.org/10.1016/0304-3975(80)90018-3) .



This *algorithms* or *data structures*-related article is a *stub*. You can help Wikipedia by *expanding it*.

v · t · e	Tree data structures	[hide]
Search trees (dynamic sets/associative arrays)	2–3 · 2–3–4 · AA · (a,b) · AVL · B · B+ · B* · B ^X · (Optimal) Binary search · Dancing · HTree · Interval · Order statistic · (Left-leaning) Red-black · Scapegoat · Splay · T · Treap · UB · Weight-balanced	
Heaps	Binary · Binomial · Fibonacci · Leftist · Pairing · Skew · Van Emde Boas	
Tries	Hash · Radix · Suffix · Ternary search · X-fast · Y-fast	
Spatial data partitioning trees	BK · BSP · Cartesian · Hilbert R · <i>k</i> -d (implicit <i>k</i> -d) · M · Metric · MIP · Octree · Priority R · Quad · R · R+ · R* · Segment · VP · X	
Other trees	Cover · Exponential · Fenwick · Finger · Fusion · Hash calendar · iDistance · K-ary · Left-child right-sibling · Link/cut · Log-structured merge · Møkle · PQ · Range · SPQR · Top	

This page was last modified on 29 August 2015, at 02:12.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

