Linux Command
Binary search trees guarantee O(h) worst-case complexity for lookup, insertion, and deletion, where h is the height of the tree. Unless care is taken, however, the height h may be as bad as N, the number of nodes. We have considered two ways of ensuring that trees stay well enough balanced so that the height, and hence the running time of operations, is O(log N). 2-3 trees require that all paths from the root to leaves are exactly the same length, but allow internal nodes to have two or three children. AVL trees require the heights of the subtrees of any node to differ by no more than one level, which ensures that the height is O(log N). Red-black trees can be viewed as an implementation of 2-3 trees that represents each 3-node as a pair of binary nodes, one red and one black. It can also be viewed as variation on AVL trees. Whereas AVL trees are binary trees with a heightfield in each node and insert and delete operations restructure the tree to ensure restrictions on heights, red-black trees are binary trees with a red bit in each node and operations that restructure (and re-color) nodes to ensure restrictions on colors. The height of an AVL tree is bounded by roughly 1.44 * $\log_2$ N, while the height of a red-black tree may be up to 2 * $\log_2$ N. Thus lookup is slightly slower on the average in red-black trees. On the other hand, it can be shown that whereas an insertion in an AVL tree may require O(log N) rotations, an insertion in a red-black tree requires only O(1) corresponding operations. O(log N) work is necessary anyhow to decide where to add the new leaf, so both kinds of tree have the same overall complexity, but the constant bound on restructuring operations in red-black trees affects certain advanced applications that are beyond the scope of this discussion.
https://gitbook.tw/chapters/using-git/working-staging-and-repository.html

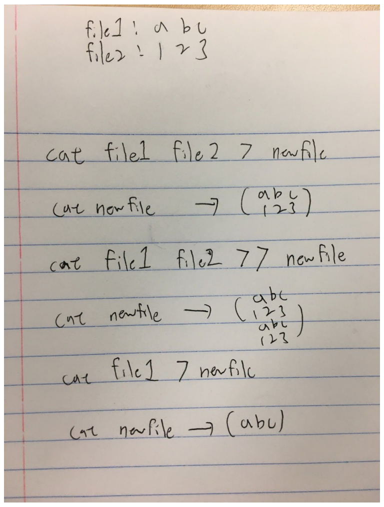
https://github.com/kevchen1129/test.git

https://www.computerhope.com/unix/ugrep.htm
https://canvas.wisc.edu/courses/121155/quizzes/86376?module_item_id=1412356

https://www.howtogeek.com/107808/how-to-manage-files-from-the-linux-terminal-11-commands-you-need-to-know/

diff:
http://www.tutorialspoint.com/articles/how-to-use-diff-command-in-linux

cat:



file1 : a b c
file2 : 1 2 3

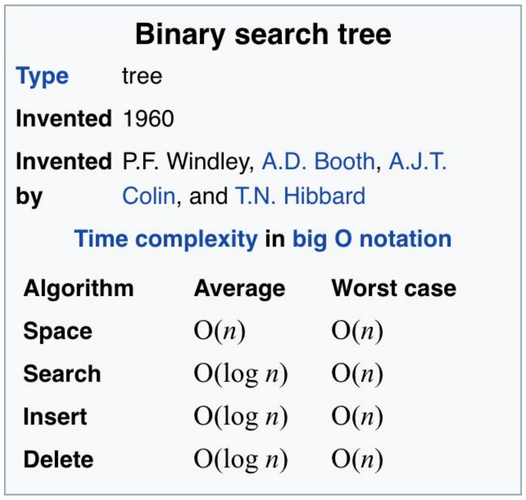
```
cat file1 file2 > newfile
cat newfile → (abc
               123)
cat file1 file2 >> newfile
cat newfile → (abc
               123
               abc
               123)
cat file1 > newfile
cat newfile → (abc)
```

Source - A node is considered a source in a graph if it has in-degree of 0 (no nodes have a source as their destination); likewise, a node is considered a sink in a graph if it has out-degree of 0 (no nodes have a sink as their source).

A **cyclic graph** is a directed graph which contains a path from at least one node back to itself.

An **acyclic graph** is a directed graph which contains absolutely no cycle, that is no node can be traversed back to itself.

complete graph: An undirected graph that has an edge between every pair of nodes

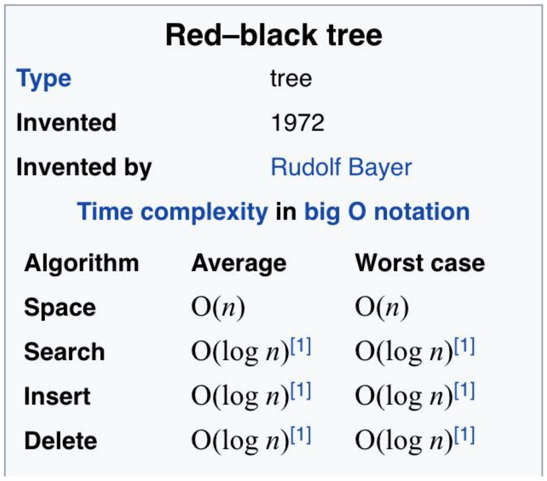


Binary search tree

Type	tree
Invented	1960
Invented by	P.F. Windley, A.D. Booth, A.J.T. Colin, and T.N. Hibbard

Time complexity in big O notation

Algorithm	Average	Worst case
Space	$\mathrm{O}(n)$	$\mathrm{O}(n)$
Search	$\mathrm{O}(\log n)$	$\mathrm{O}(n)$
Insert	$\mathrm{O}(\log n)$	$\mathrm{O}(n)$
Delete	$\mathrm{O}(\log n)$	$\mathrm{O}(n)$

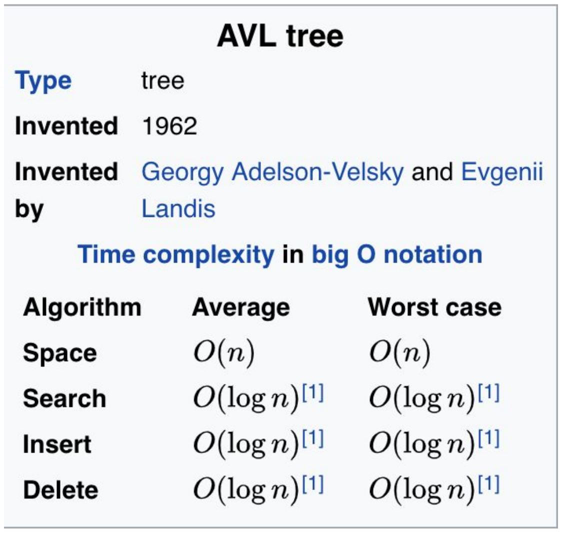


Red–black tree

Type	tree
Invented	1972
Invented by	Rudolf Bayer

Time complexity in big O notation

Algorithm	Average	Worst case
Space	$\mathrm{O}(n)$	$\mathrm{O}(n)$
Search	$\mathrm{O}(\log n)$[1]	$\mathrm{O}(\log n)$[1]
Insert	$\mathrm{O}(\log n)$[1]	$\mathrm{O}(\log n)$[1]
Delete	$\mathrm{O}(\log n)$[1]	$\mathrm{O}(\log n)$[1]

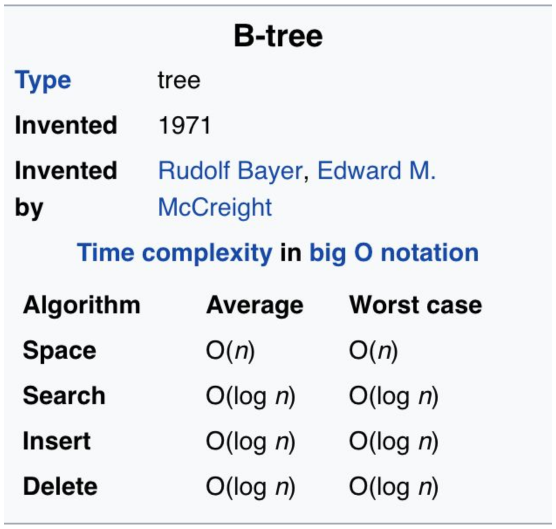


AVL tree

Type	tree
Invented	1962
Invented by	Georgy Adelson-Velsky and Evgenii Landis

Time complexity in big O notation

Algorithm	Average	Worst case
Space	$O(n)$	$O(n)$
Search	$O(\log n)$[1]	$O(\log n)$[1]
Insert	$O(\log n)$[1]	$O(\log n)$[1]
Delete	$O(\log n)$[1]	$O(\log n)$[1]

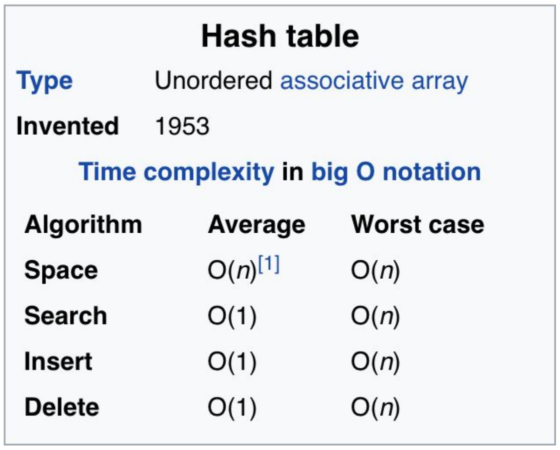


B-tree

Type	tree
Invented	1971
Invented by	Rudolf Bayer, Edward M. McCreight

Time complexity in big O notation

Algorithm	Average	Worst case
Space	O(n)	O(n)
Search	O(log n)	O(log n)
Insert	O(log n)	O(log n)
Delete	O(log n)	O(log n)



Hash table

Type	Unordered associative array
Invented	1953

Time complexity in big O notation

Algorithm	Average	Worst case
Space	O(n)[1]	O(n)
Search	O(1)	O(n)
Insert	O(1)	O(n)
Delete	O(1)	O(n)