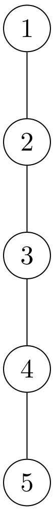
Trie

Notation : `trie`

Description : A structure for organizing sequential data hierarchically. Members of a previous generation can spawn infinitely many members of the next generation, but the data cannot be read out of sequence. In other words a parent can spawn any number of children, but cannot skip a generation. This ensures that all the data in a trie are related to each other properly.

A simple kind of trie that everyone has used is a list:



1 Trie Terminology[1]

Root – The top (first) node in a tree.

Child – A node directly connected to another node when moving away from the Root.

Parent – The converse notion of a child.

Siblings – A group of nodes with the same parent.

Descendant – A node reachable by repeated proceeding from parent to child.

Ancestor – A node reachable by repeated proceeding from child to parent.

Leaf [1]– A node with no children.

Branch[2] – A node with at least one child.

Degree – The number of subtrees of a node.

Edge – The connection between one node and another.

Path – A sequence of nodes and edges connecting a node with a descendant.

Level – The level of a node is defined by 1 + (the number of connections between the node and the root).

Node Height – The height of a node is the number of edges on the longest path between that node and a leaf.

Tree Height – The height of a tree is the height of its root node.

Depth – The depth of a node is the number of edges from the tree's root node to the node.

Forest – A forest is a set of $n \geq 0$ disjoint trees.

References

[1] W. contributors, *Tree (data structure) — wikipedia, the free encyclopedia*, [Online; accessed 15-December-2017], 2017. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Tree_(data_structure)&oldid=813972413`.

[1] a.k.a. External Node