

Trees

Definition

Unlike Lists, Stacks and Queues which are linear data structures, a tree is one of the non-linear data structures. It is recursively defined as a set of nodes such that it either:

- is an empty set of nodes, or
- has a root node from which 0 or more trees descend

Implementation

A tree can be implemented recursively (co-recursively to be exact, start from the base case and expand) using list of lists. The first element is the root, the second element is the left sub-tree and the third element is the right sub-tree. Another approach is to use nodes and references to better follow object-oriented programming principles. One of such representation is called *First Child/Next Sibling* representation. In such representation, each node has 2 pointers:

- One to its *first* child
- Second to its *next* sibling

The benefit of such representation is that it allows arbitrary number of children. The implementation *tree.py* uses this representation.

Traversal

There are three ways a tree can be traversed in Depth First Search manner (considering left-to-right traversal).

- Pre-order Traversal: root node \rightarrow left sub-tree \rightarrow right sub-tree
- Post-order Traversal: left sub-tree \rightarrow right sub-tree \rightarrow root node
- In-order Traversal: left sub-tree \rightarrow root node \rightarrow right sub-tree

Tree can also be traversed in Breadth First Search manner.

- Level-order Traversal: level 0 \rightarrow level 1 \rightarrow level 2 \rightarrow ...