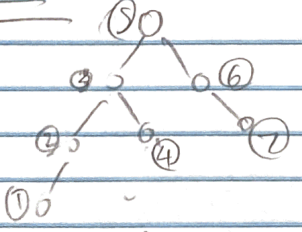
Binary Trees

Structure info:
~~Three pointers~~ per node, one value
- item, parent, left, right

Definitions:
root: node on top, no parent
leaf: node on bottom layer, no children
depth: length of path from root to lowest child
height: max depth

Traversal
traversal is done from left to right. If written recursively, then
traverse tree {
    traverse left tree;
    traverse right tree;
}
is a recursive algorithm for defining it.

Example of traversal:



Starts from bottom.

Results:
- Height of tree is $\log_2(n)$ time of n.
- Traversal of tree is $\log_2(n) \approx n$ time

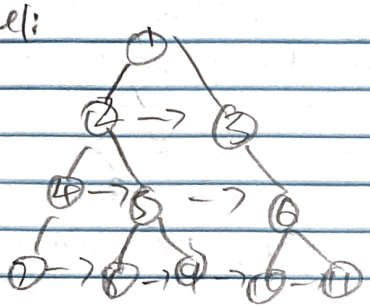
The two main ways of traversal are:
BFS
DFS.

BFS (Breadth First Search):

Steps:

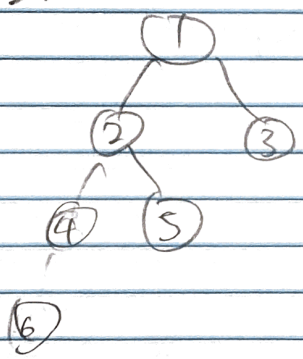
1. Visit/mark starter node
2. Visit/mark all away from starter node
3. Continue to do so, from left to right

Example1:



Order of traversal:

Example 2:



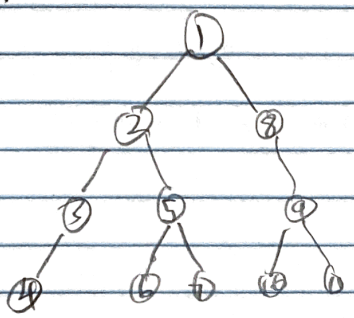
Literally layer by layer, with runtime $O(\log_2 n)$

Depth first search:

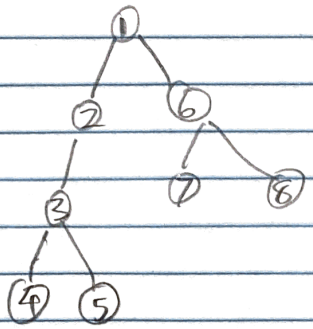
Steps:

1. mark root, and traverse left
2. If nothing left to marke, return to most recent node with an marked right child and start there
3. repeat.

Example1:



Example2:



Runtime of DFS is also $O(\log_2 n)$.

Source: MIT OCW Lecture 6.006.