Binary Trees & Algorithms

Oluwatimi Owoturo's summary

Concept

Definition: A binary tree is a data structure which consists of 6 parts. We have a root, internal node, external node, ancestors, descendants and subtrees.

6 PARTS

Root: Take the root of a tree. This root has nothing below it (well except dirt) but in a binary tree, the root is at the top (flip a tree, and you have a binary tree). Well, in programming terms it is a node without a parent.

Internal node: This would be a node with at least one child (let's say a branch that has at least on leaf)

External node: This is a node without children (This would be a leaf that has nothing after itself (always last))

Ancestors of a node: Parent, grandparent of a node. (This is what comes before a node) a root has no Ancestor.

Descendant of a node: This is a child, grandchild or subtree.

Subtree: A subpart of a binary tree that consists of a node and its descendants

Other important concepts

Distance between two nodes: # of edges between them

Depth of a node: Number of ancestors (distance or # nodes from the root)

Height of a tree: Maximum depth of any node.

Formulas

E = number of leaves |N = number of nodes.

I = number of internal nodes (non-leaves) | H = height.

$$e = i + 1$$

$$n = 2e - 1$$

$$i = \frac{n - 1}{2}$$

$h = floor(log_2(n))$ | max nodes at a level = 2i