Introduction to Trees

Ian Ludden

Learning Objectives

By the end of this lesson, you will be able to:

Learning Objectives

By the end of this lesson, you will be able to:

- Define and use tree terminology.

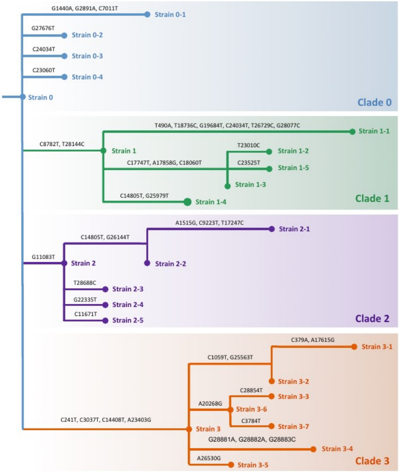
Learning Objectives

By the end of this lesson, you will be able to:

- Define and use tree terminology.
- Define and identify various tree properties.

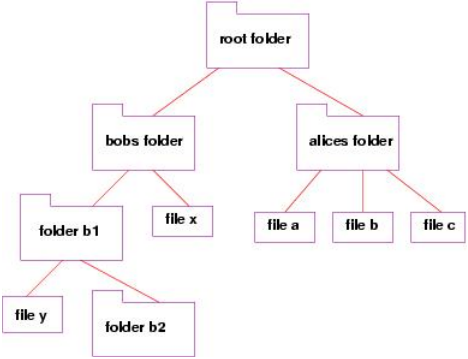
Why do we care about trees?

Why do we care about trees?



A phylogenetic tree (Source)

Why do we care about trees?



A file tree (Source)

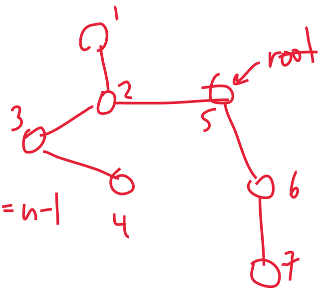
What is a tree?

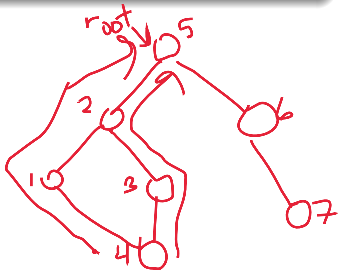
Definition

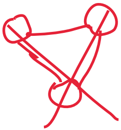
A ***tree*** is a connected acyclic graph. A ***rooted*** tree has a special vertex called a ***root***.

$T = (V, E)$ ← nodes

$|V| = n \rightarrow |E| = n-1$







What is a tree?

Definition

A ***tree*** is a connected acyclic graph. A ***rooted*** tree has a special vertex called a ***root***.

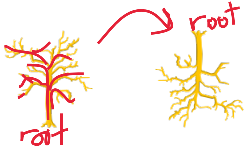
- Root goes at the top by convention

What is a tree?

Definition

A ***tree*** is a connected acyclic graph. A ***rooted*** tree has a special vertex called a ***root***.

- Root goes at the top by convention
- Terms borrowed from biological trees and family trees



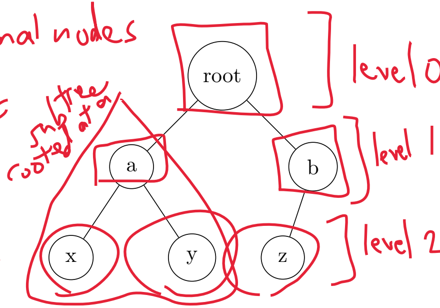
What is a tree?

Definition

A ***tree*** is a connected acyclic graph. A ***rooted*** tree has a special vertex called a ***root***.

- Root goes at the top by convention
- Terms borrowed from biological trees and family trees

internal nodes



leaf nodes

leaves

parent/child/sibling

ancestor/descendant "proper"

height := max level

m-ary trees

Definition

An ***m*-ary tree** is a tree in which each node has at most *m* children.

m=2: binary tree
m=3: ternary tree

m-ary trees

Definition

An ***m-ary tree*** is a tree in which each node has at most m children.

Some special cases (shown for $m = 2$):

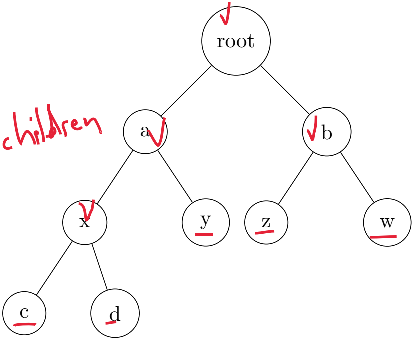
Definition

An ***m*-ary tree** is a tree in which each node has at most m children.

Some special cases (shown for $m = 2$):

- ***full*** m-ary tree

every node: 0 or m children



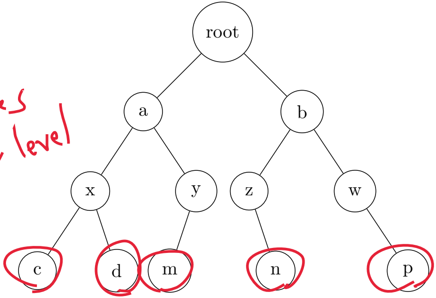
Definition

An ***m*-ary tree** is a tree in which each node has at most *m* children.

Some special cases (shown for $m = 2$):

- ***full*** *m*-ary tree
- ***complete*** *m*-ary tree

all leaves at same level

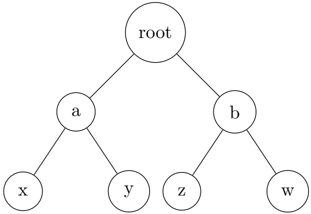


Definition

An ***m*-ary tree** is a tree in which each node has at most m children.

Some special cases (shown for $m = 2$):

- ***full*** m-ary tree
- ***complete*** m-ary tree
- full *and* complete m-ary tree



Counting Nodes

Counting Nodes

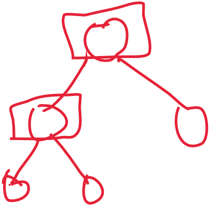
Fact

A full m-ary tree with i internal nodes has $mi + 1$ nodes total.

Proof: Ask everyone how many kids they have (then add the root).

Internal nodes: m each
Leaf nodes: 0 each

$mi + 1$



Counting Nodes

Fact

A full m-ary tree with i internal nodes has $mi + 1$ nodes total.

Proof: Ask everyone how many kids they have (then add the root).

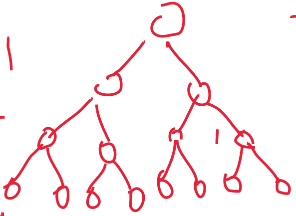
Fact

A binary tree of height h has at least $h + 1$ and at most $2^{h+1} - 1$ nodes.

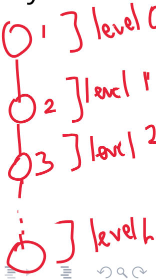
Proof: Consider a path of length h and a full, complete binary tree of height h.

$$\sum_{k=0}^{h} 2^k = 2^{h+1} - 1$$

$$\sum_{k=0}^{h} m^k = \frac{m^{h+1} - 1}{m-1}$$



level	# nodes
0	1
1	2
2	4
3	8
k	2^k



Counting Nodes

Fact

A full m-ary tree with i internal nodes has $mi + 1$ nodes total.

Proof: Ask everyone how many kids they have (then add the root).

Fact

A binary tree of height h has at least $h + 1$ and at most $2^{h+1} - 1$ nodes.

Proof: Consider a path of length h and a full, complete binary tree of height h.

Fact

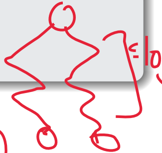
The height of a full and complete binary tree with n nodes is proportional to $\log_2 n$.

$n = 2^{h+1} - 1$

[a, b, c, d, e, f] $O(n)$

$\log_2(n+1) = h+1$

$h = \log_2(n+1) - 1 \approx \log_2(n)$



Recap: Learning Objectives

By the end of this lesson, you will be able to:

- Define and use tree terminology.
- Define and identify various tree properties.