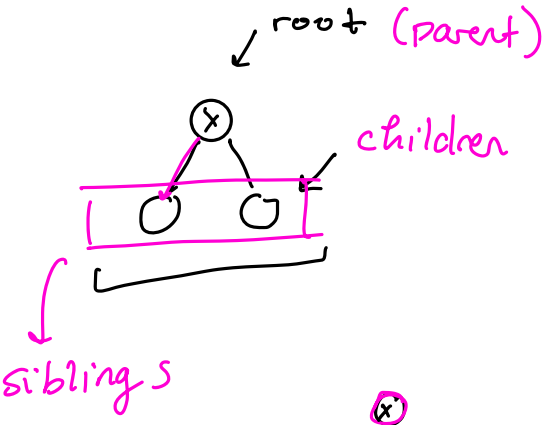
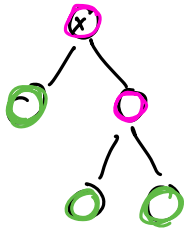
Trees:

- undirected graphs
- 1 root node
- finite # of nodes

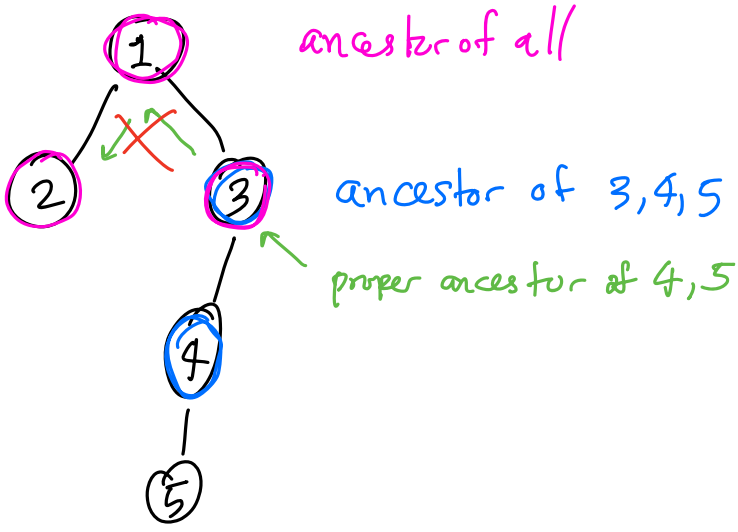


Leaves: nodes at bottom w/ no children



internal node: not leaf, has children

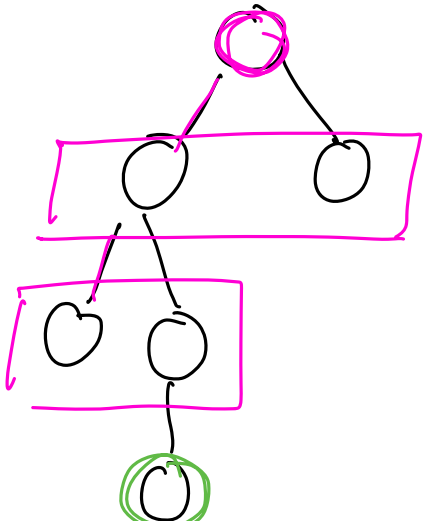
ancestor / descendent nodes:



Structure

level: # of edges to get back to root node

height: max level of any node



Level
0
1
2
3

Types of trees:

m-ary trees: a tree that can have between 0 & m childre

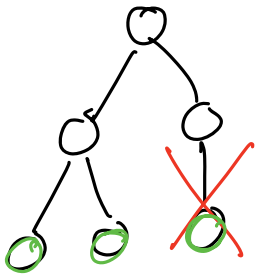
↓

branching factor

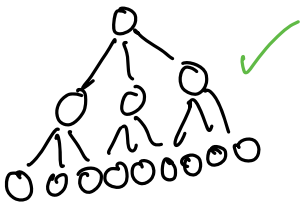
full m-ary trees: either 0 or m children

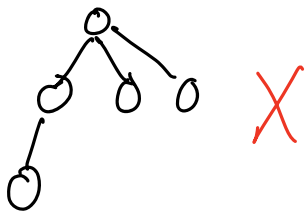
↓ ↓

leaf node internal

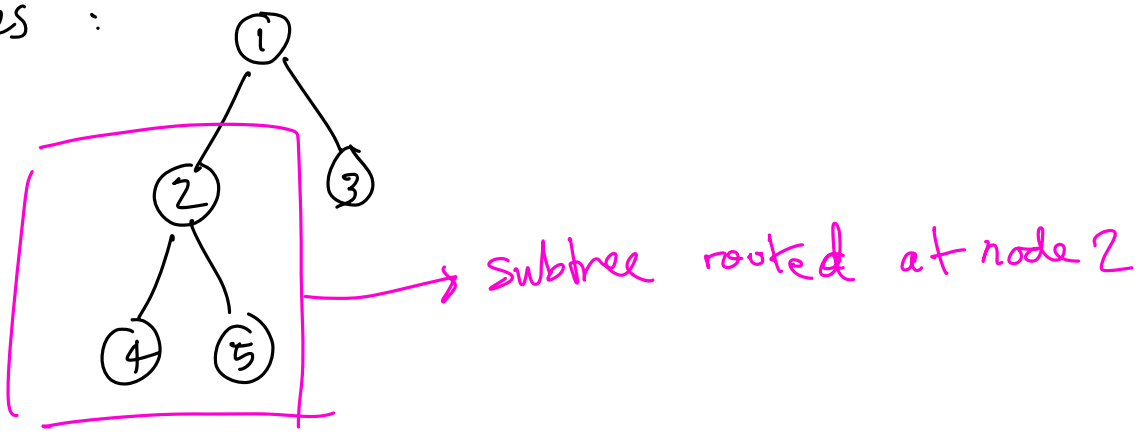


complete: all leaves at same level





subtrees:

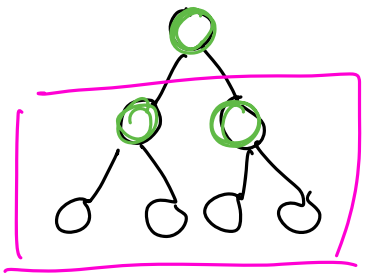


Counting nodes:

given a tree with i internal nodes, how many total nodes?

↓

full & complete m-ary tree



i) # nodes with no parent: root, 1

ii) # children nodes : # parents × branching factor

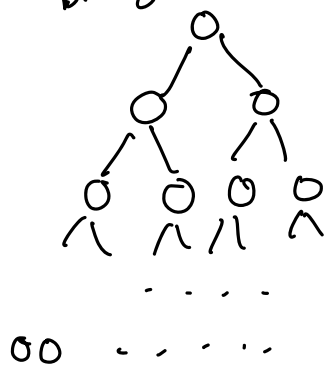
$i \times m$

$$n = mi + 1$$

$$\#\text{leaves} = n - i$$

e.g.,
full & complete n-ary tree of height h, how many nodes?

binary tree example



LVL	#nodes
0	$1 = 2^0$
1	$2 = 2^1$
2	$4 = 2^2$
...	...
h	2^h ←

$\frac{2^h}{(2^{h+1}-1)}$

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

$\log_2(2^{h+1}) = \log_2(n)$

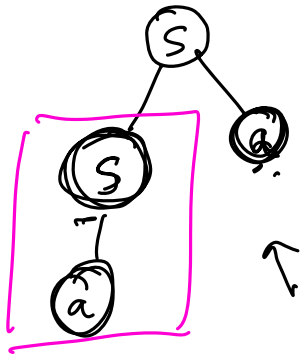
$$h \approx \log_2(n)$$

$h \approx \log_2(n)$

balanced: tree is approx. complete $h \approx \log_2(n)$

Context-free grammar (CFG):

- set of rules for structure of a tree



$S \rightarrow Sa$

$S \rightarrow a$

CFG rules

non-terminal

↳ all possible children for that rule

terminal sequence: aa

e.g. create a grammar to generate strings that are any # of "a"s (at least 1)

a^+

e.g. a, aa, aaaaaa...

another grammar:

start: S

terminals: a, b, c

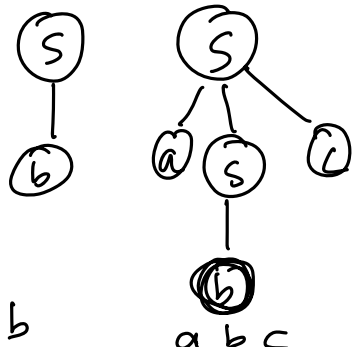
~~a^*bc^*~~ a^nbc^n | * : 0 or more

$S \rightarrow b$

$S \rightarrow aSc$

↳ any number of "a", 1 b, ↳ same # of "c"

a^nbc^n



add these rules for a^*bc^*

$S \rightarrow aS$

$S \rightarrow Sc$

want: $a^* b^*$

epsilon

$$S \rightarrow \varepsilon \mid aS \mid Sb$$