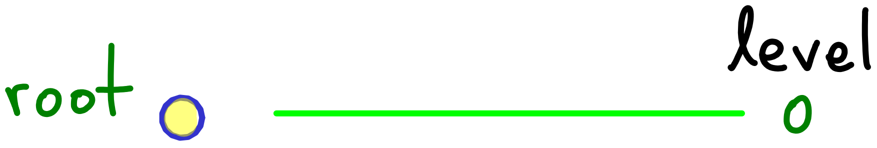
BINARY TREES

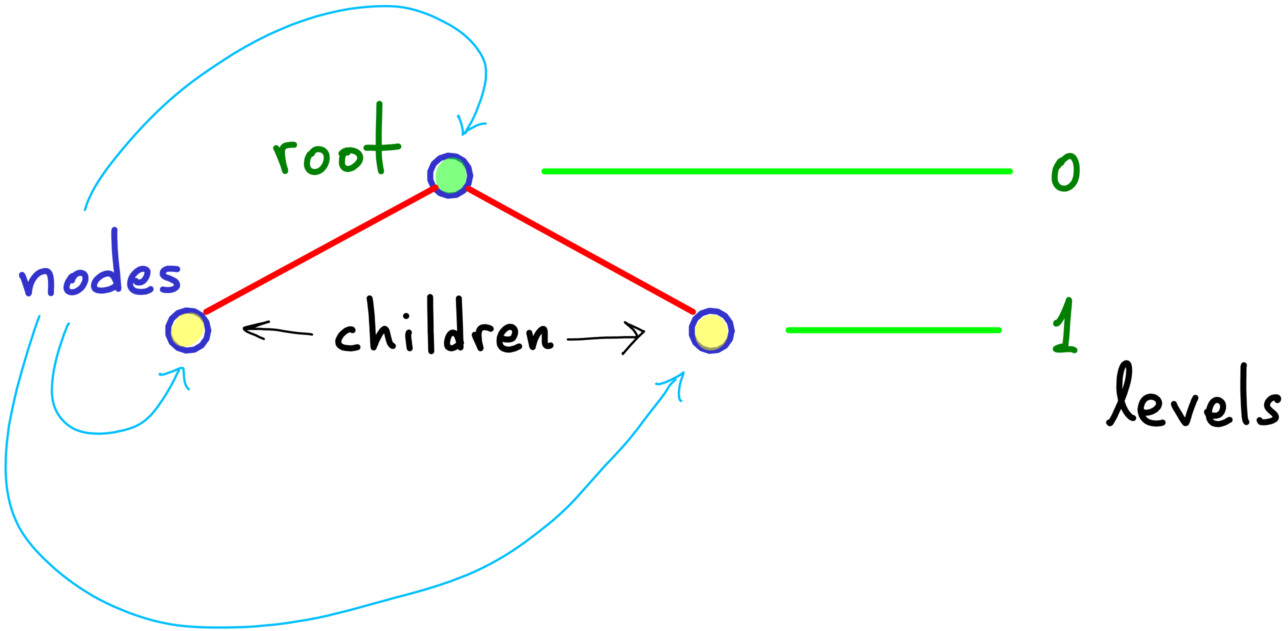
BINARY TREES

- if non-empty, $\exists$ root at level 0.



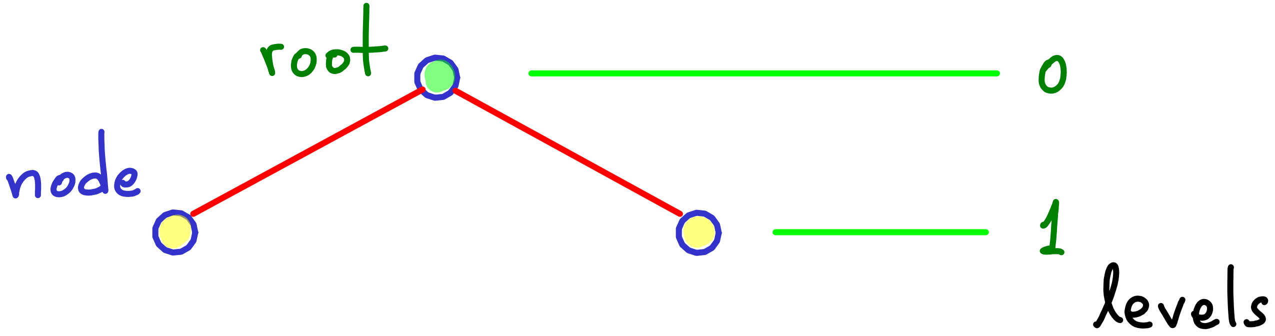
BINARY TREES

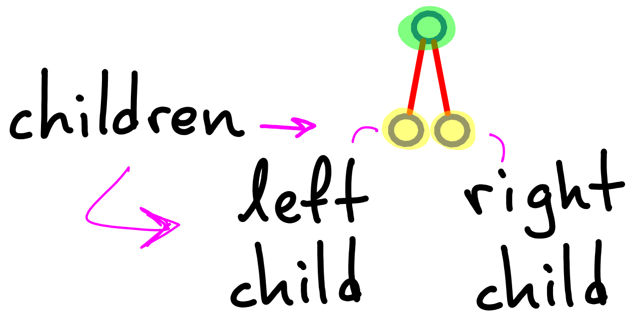
- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
  (nodes)



BINARY TREES

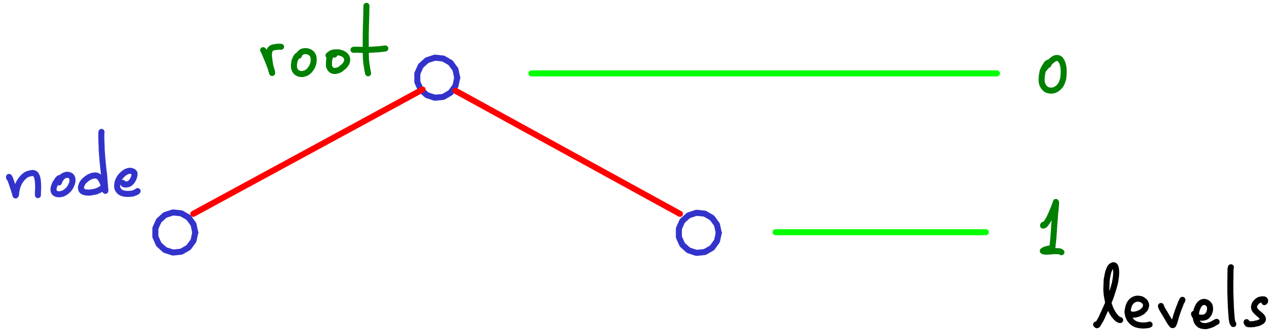
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.

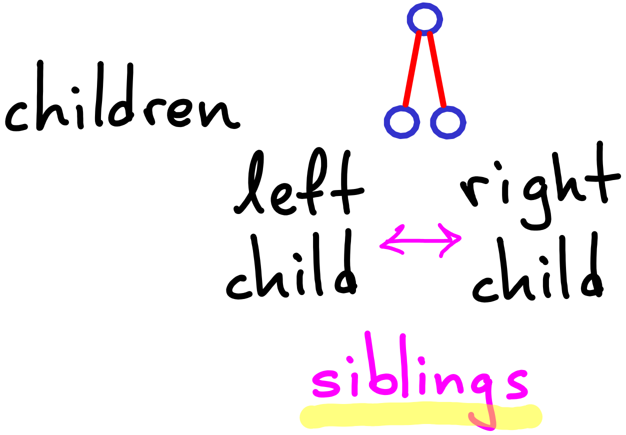




BINARY TREES

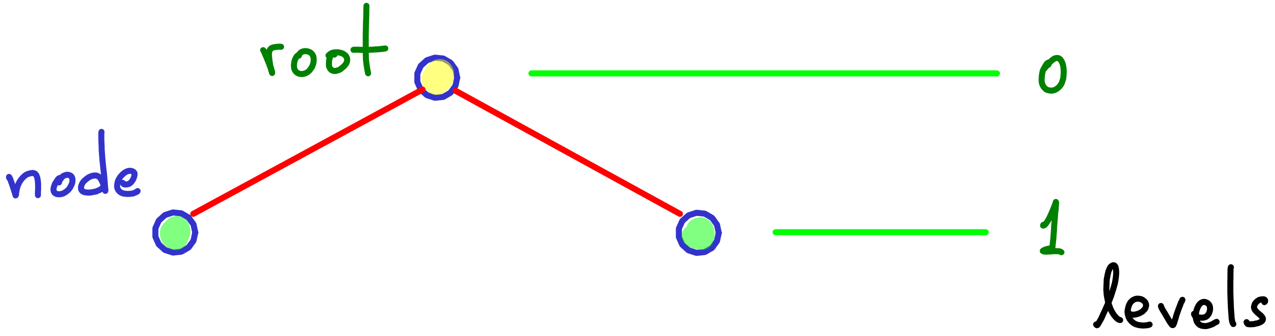
- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.

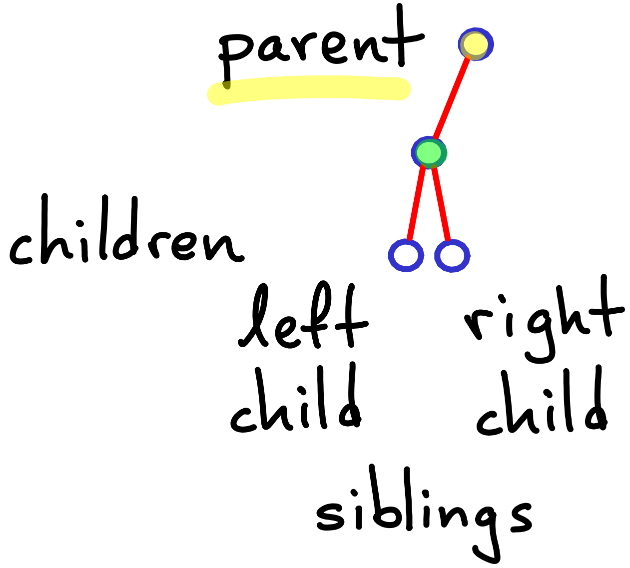




BINARY TREES

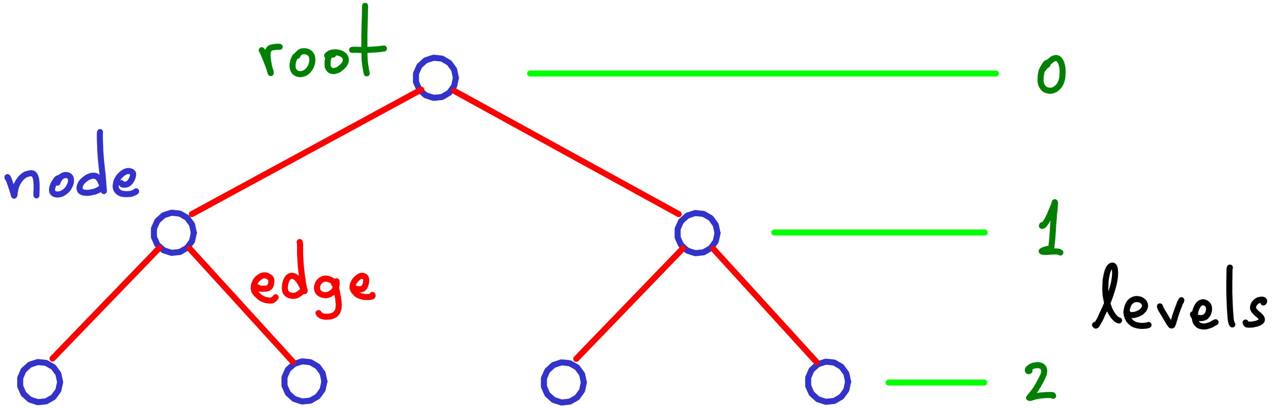
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

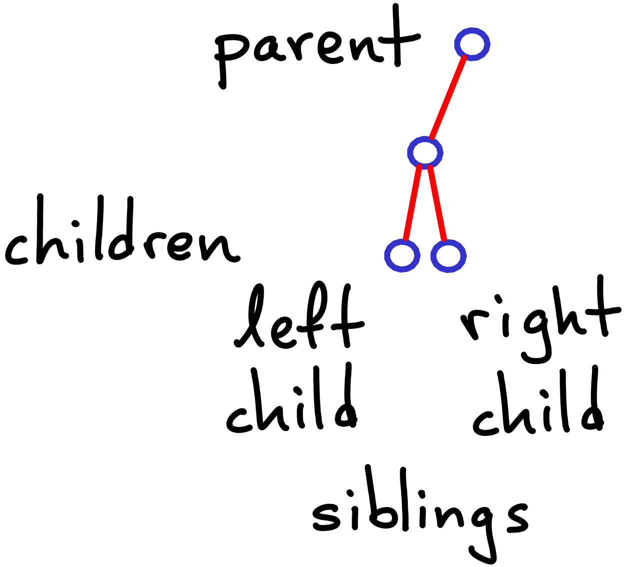




BINARY TREES

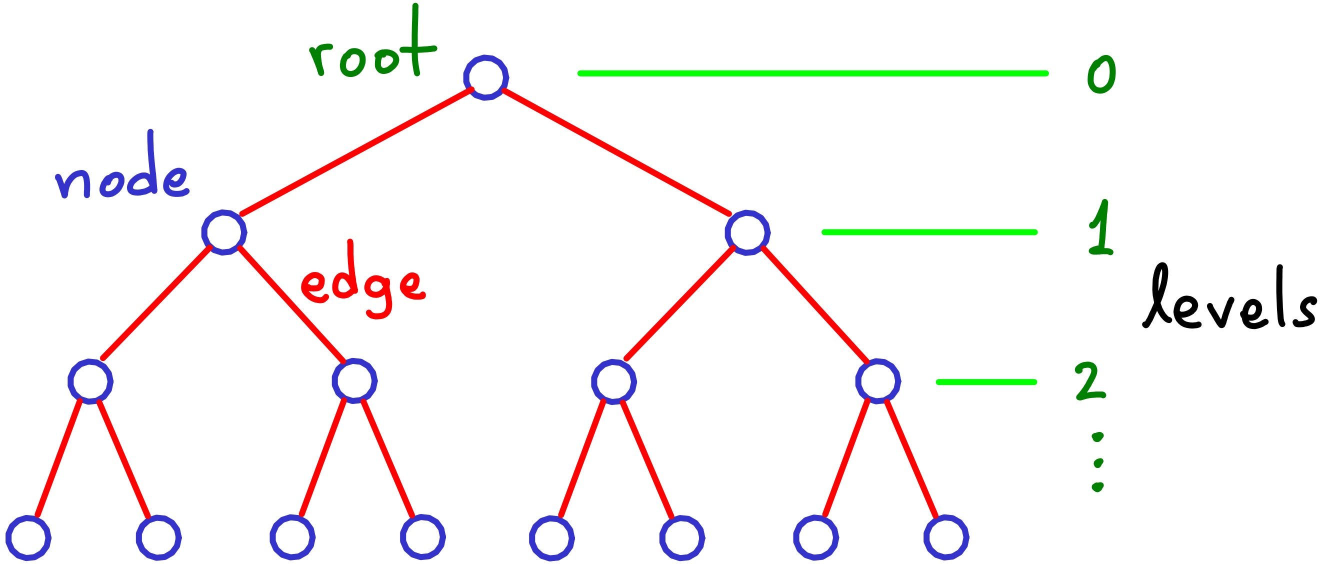
- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

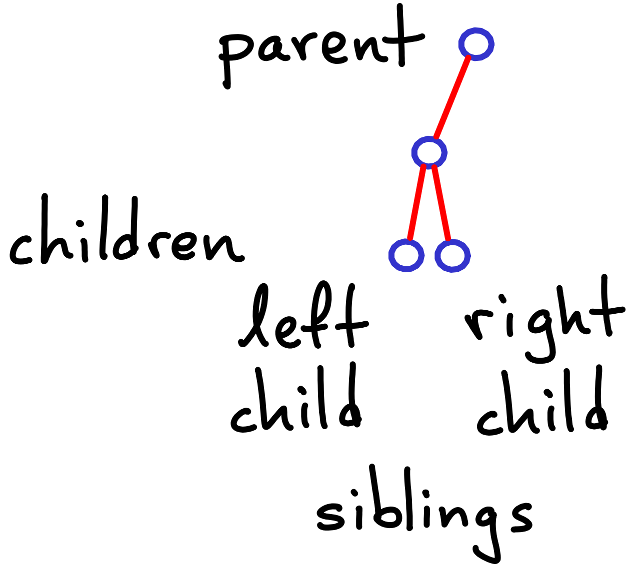




BINARY TREES

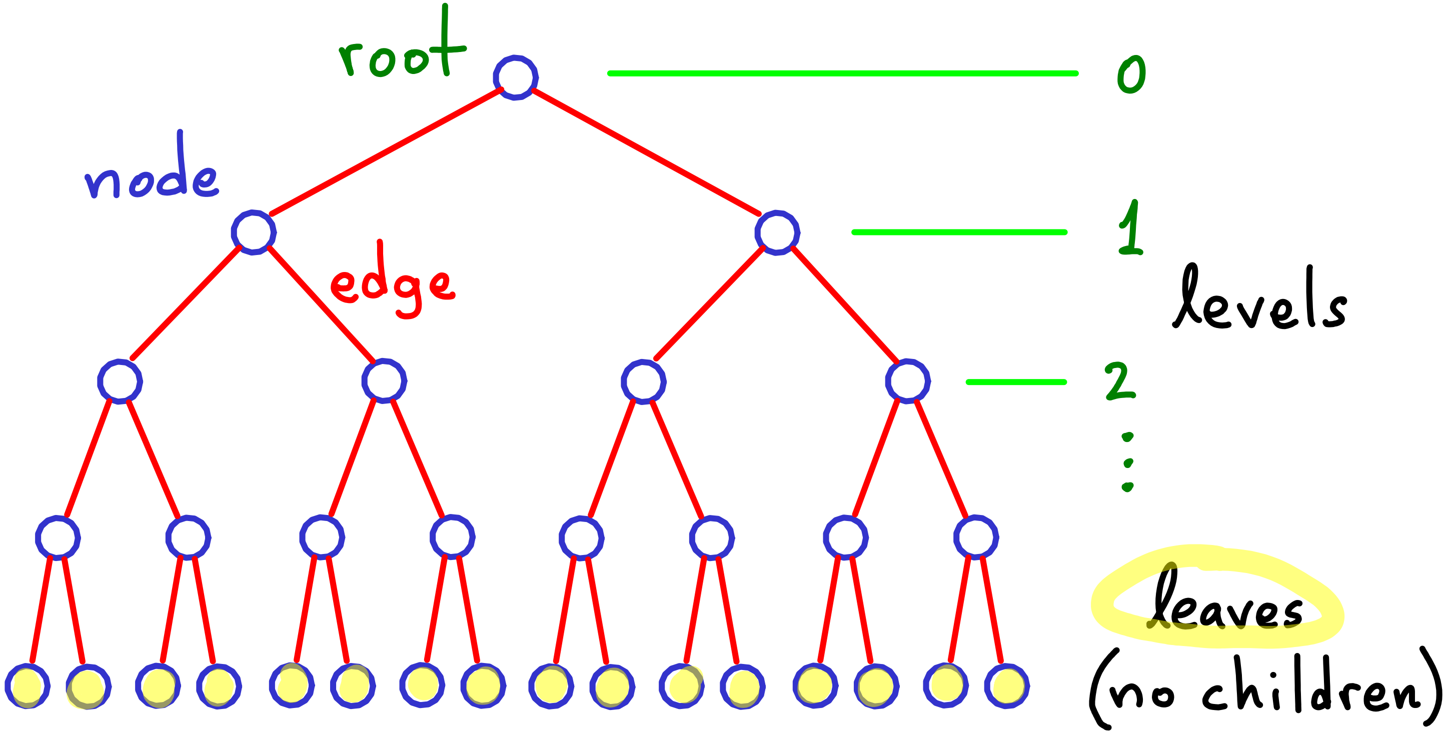
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

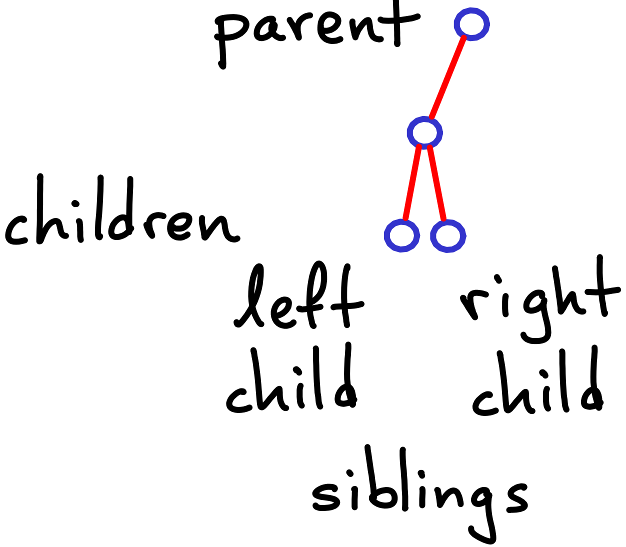




BINARY TREES

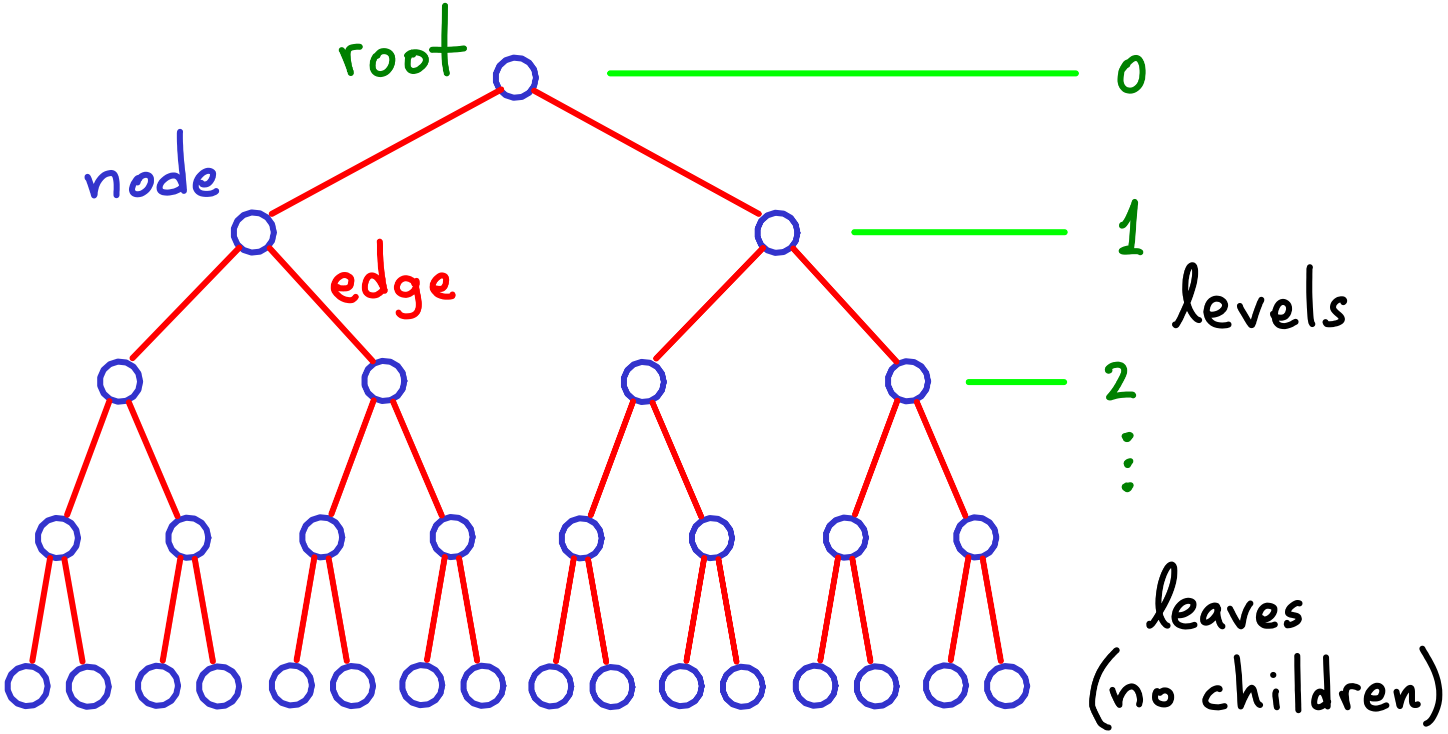
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

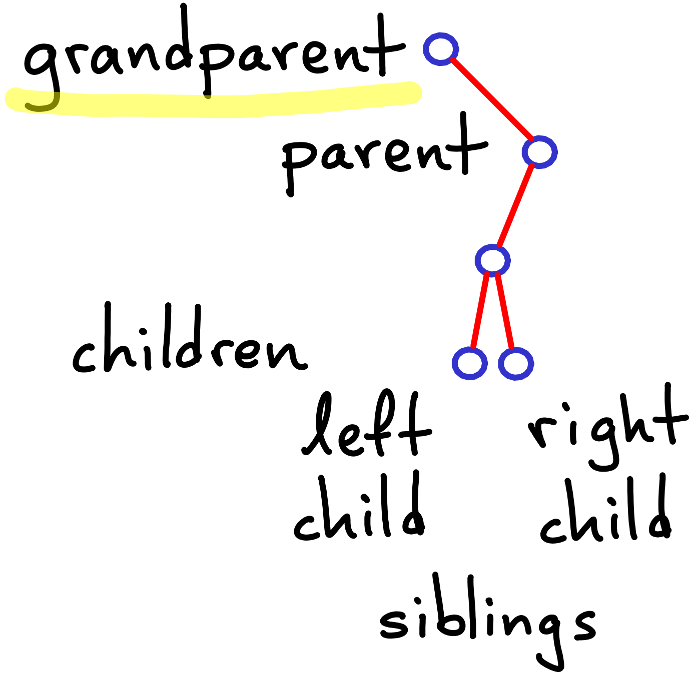




BINARY TREES

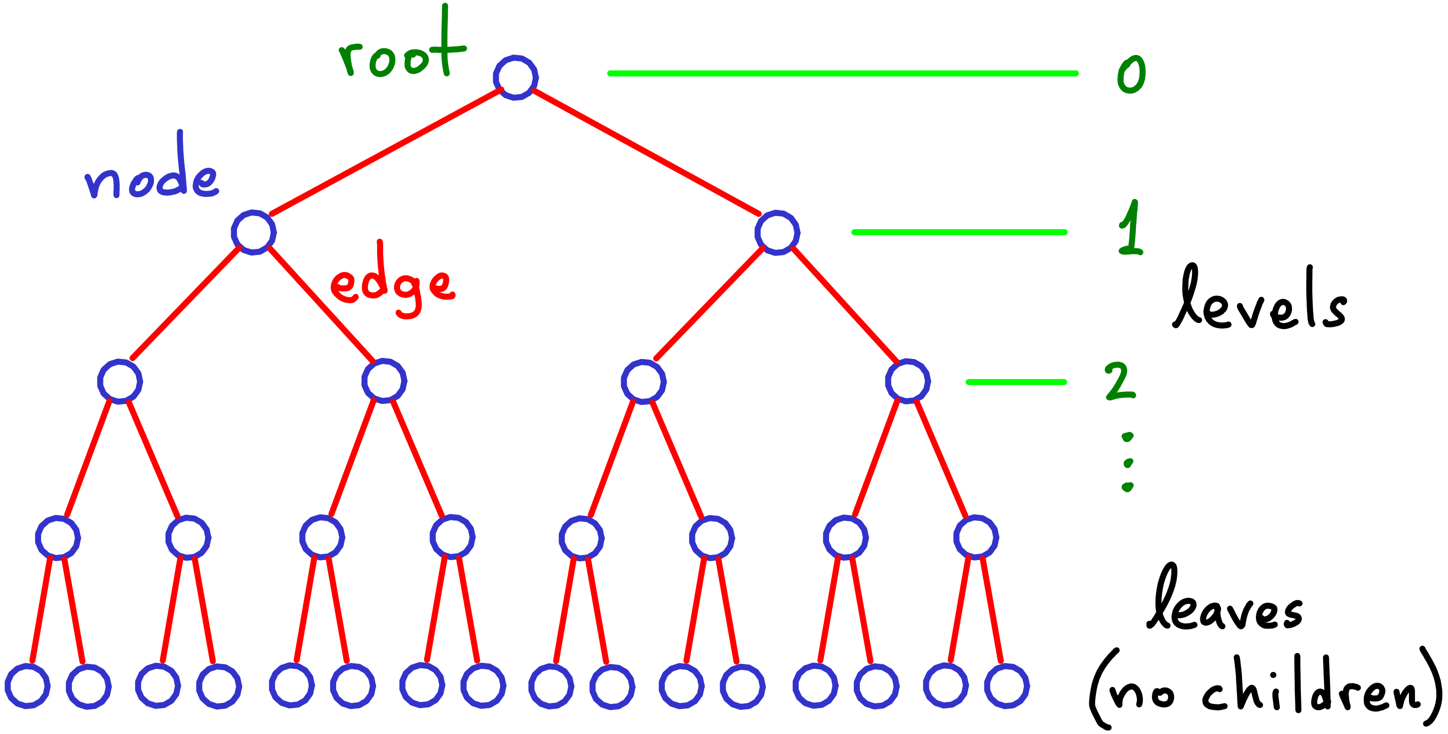
- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

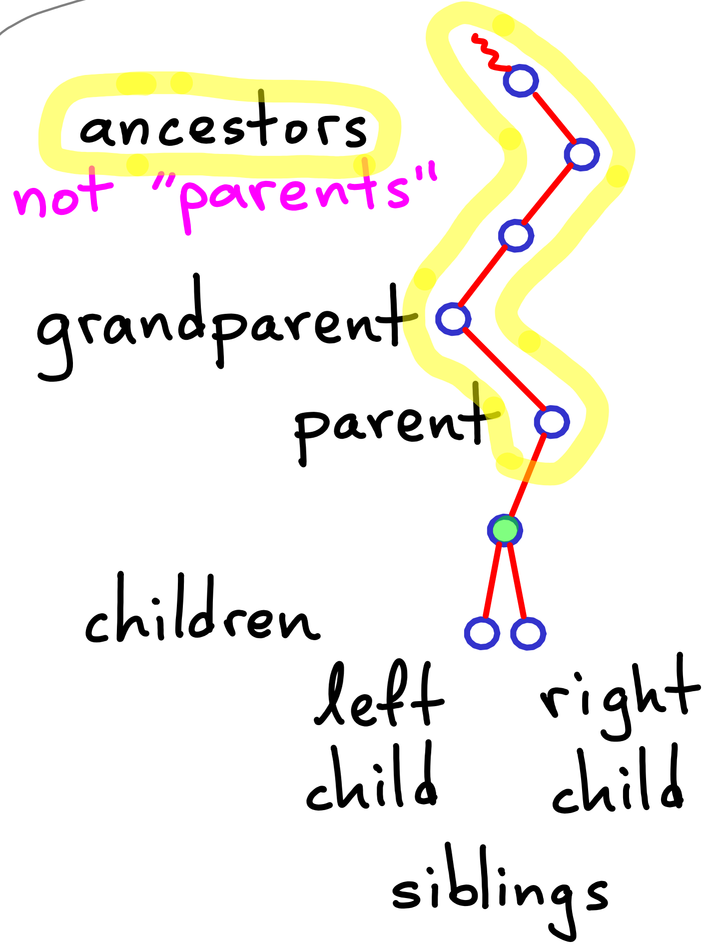




BINARY TREES

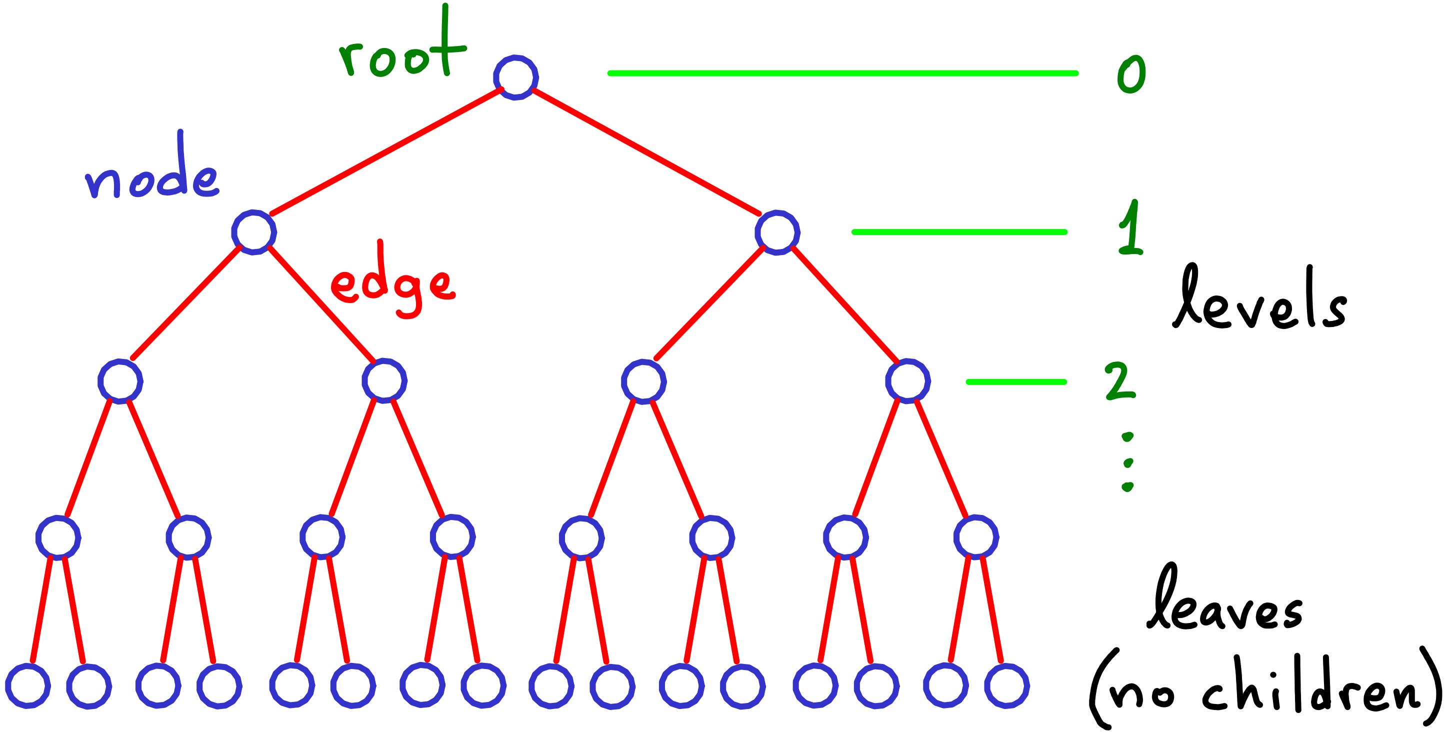
- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

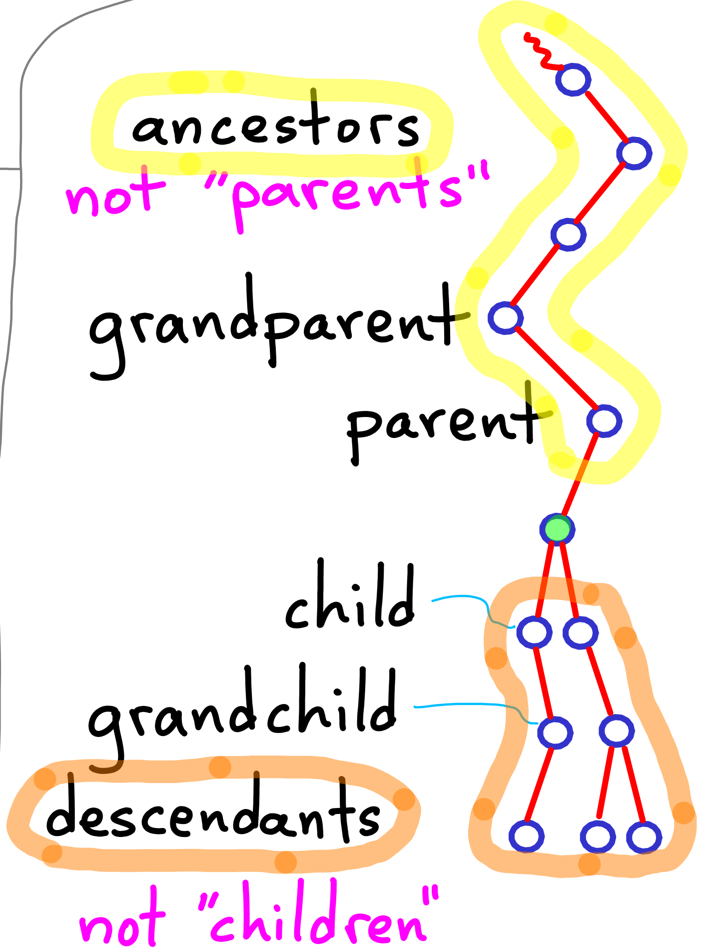




BINARY TREES

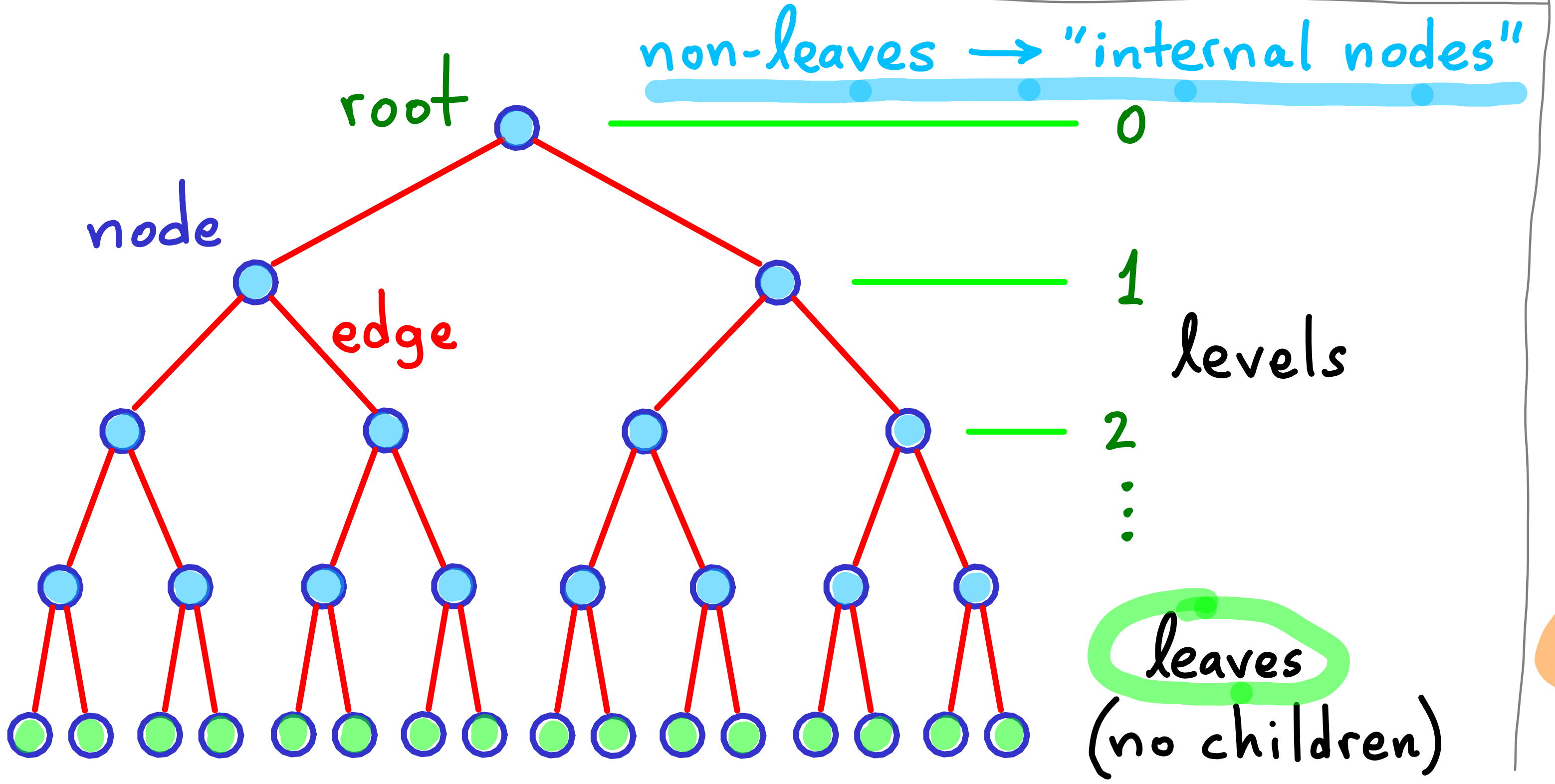
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent

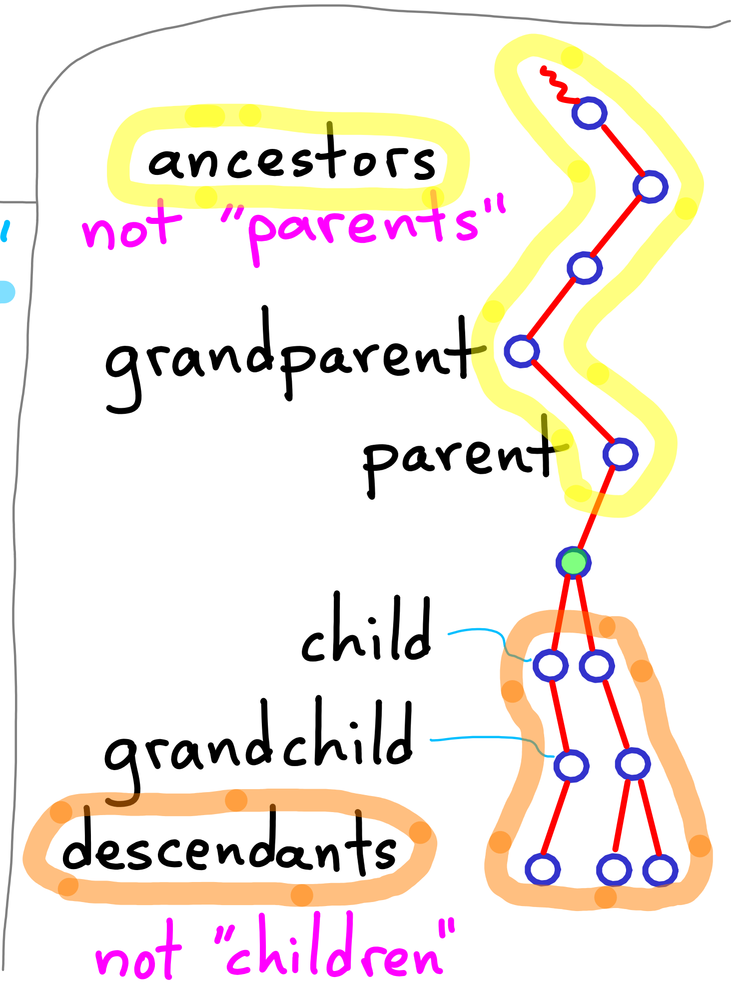




BINARY TREES

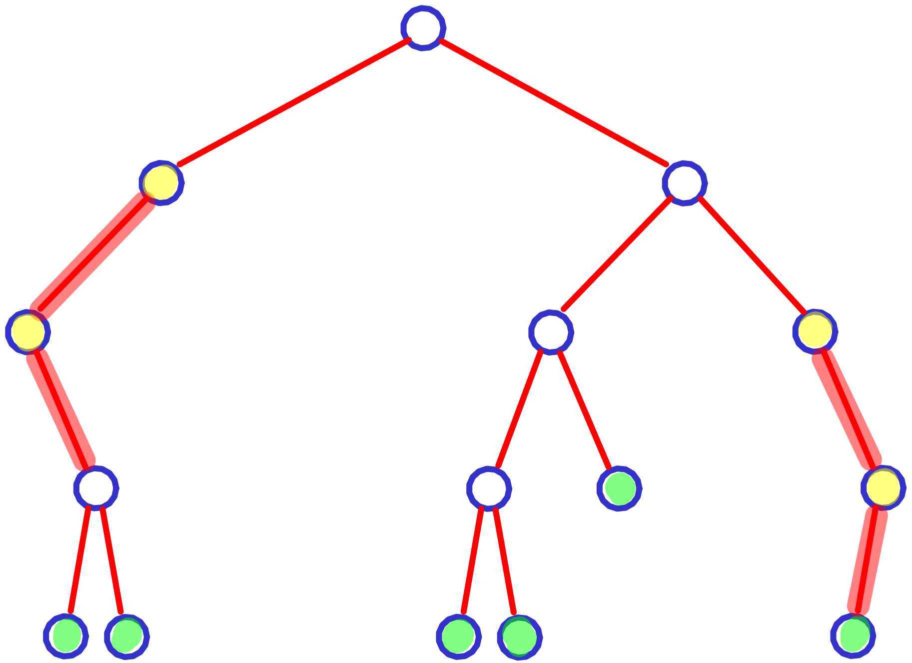
- if non-empty, ∃ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent





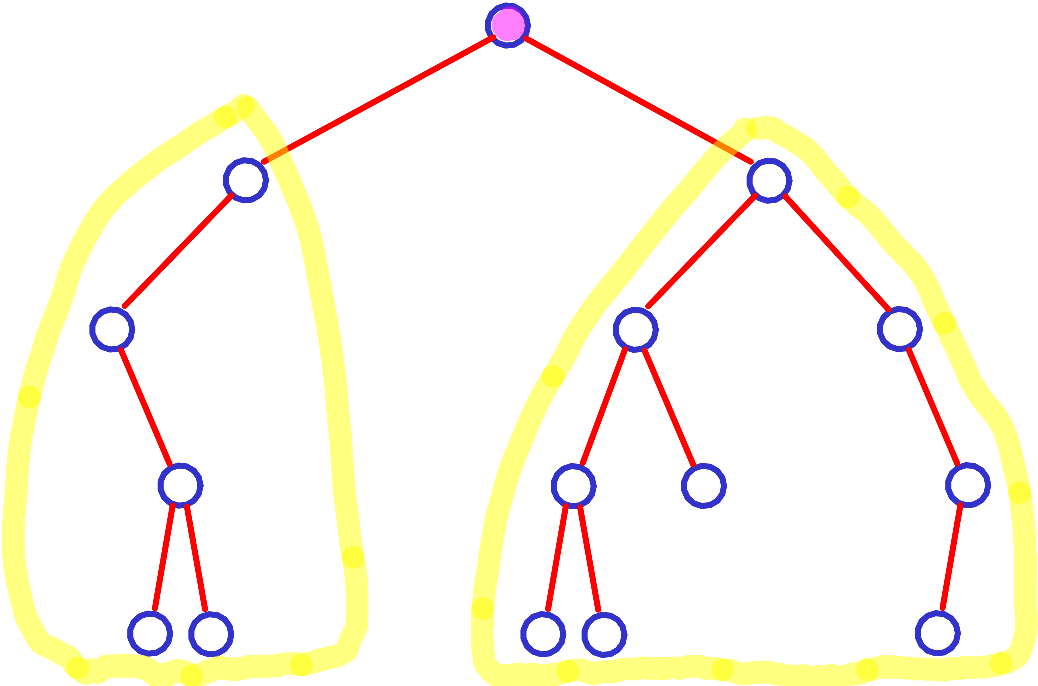
BINARY TREES

- if non-empty, $\exists$ root at level 0.
- every node at level i connects to ≤ 2 children at level $i+1$.
- every non-root node has 1 parent



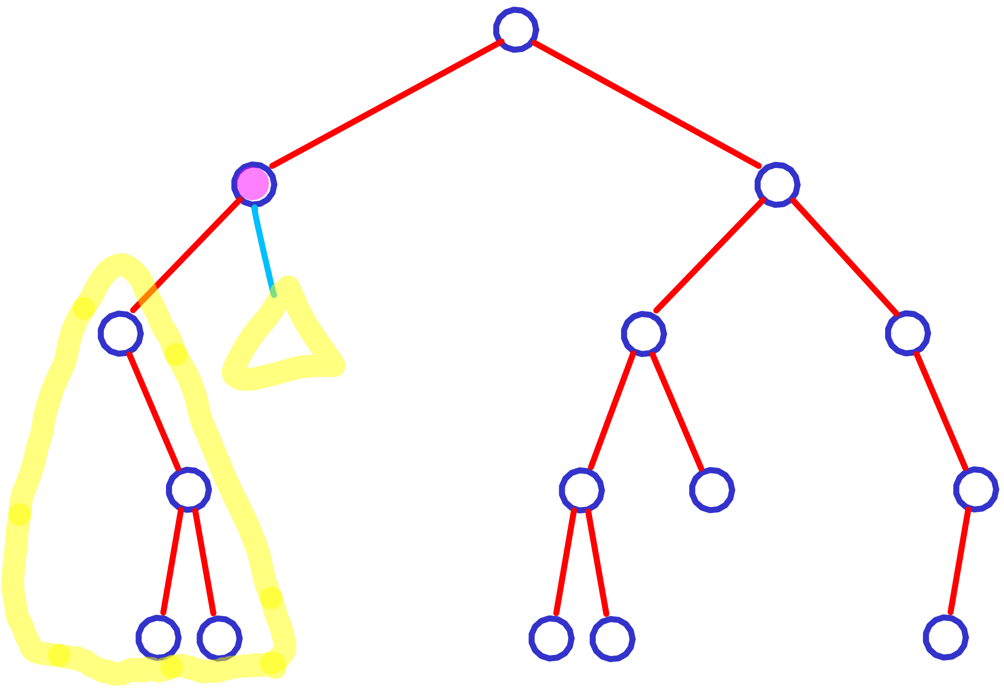
BINARY TREES

- if non-empty, ∃ root at level 0.
- every node at level i connects to 2 subtrees at level $i+1$.



BINARY TREES

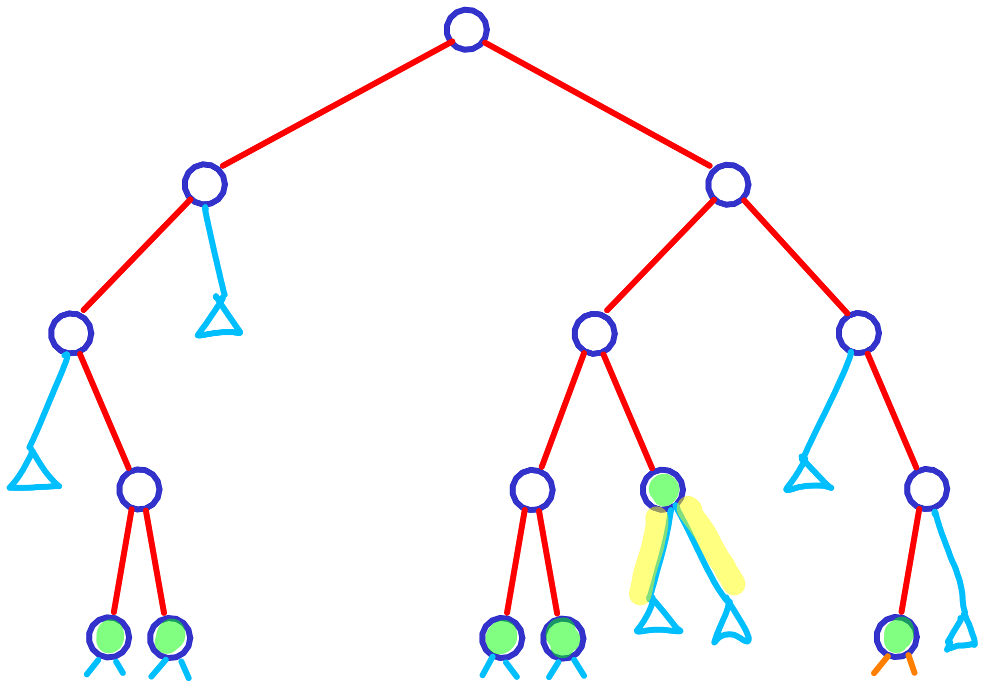
- if non-empty, ∃ root at level 0.
- every node at level i connects to 2 subtrees at level $i+1$.



[subtrees can be empty]

BINARY TREES

- if non-empty, ∃ root at level 0.
- every node at level i connects to 2 subtrees at level $i+1$.

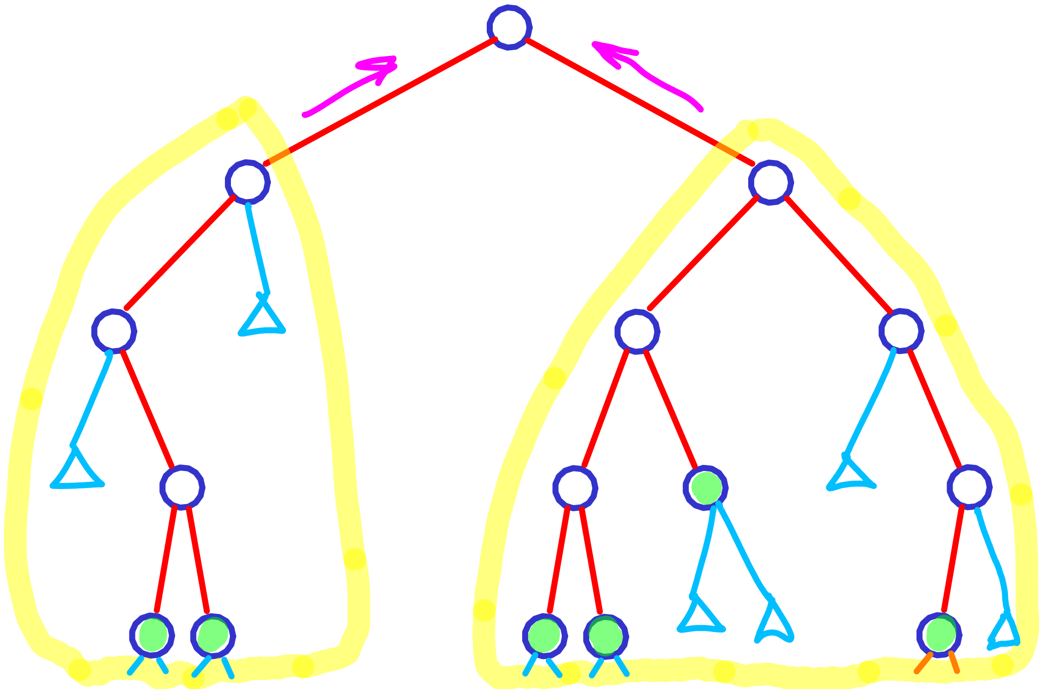


[subtrees can be empty]

leaves : 2 empty subtrees

BINARY TREES

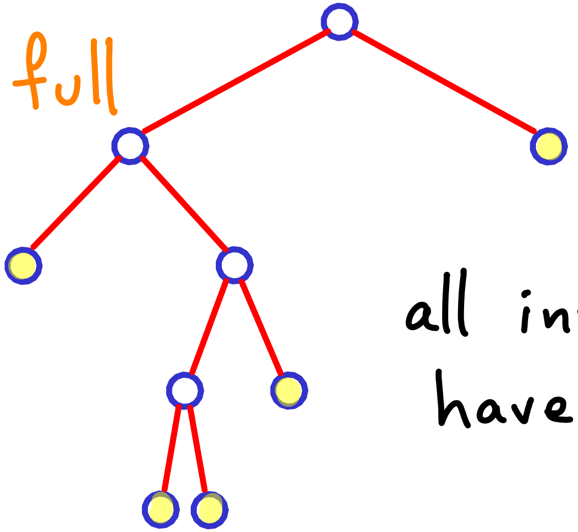
- if non-empty, ∃ root at level 0.
- every node at level i connects to 2 subtrees at level $i+1$.
- every subtree root has 1 parent (except global root)



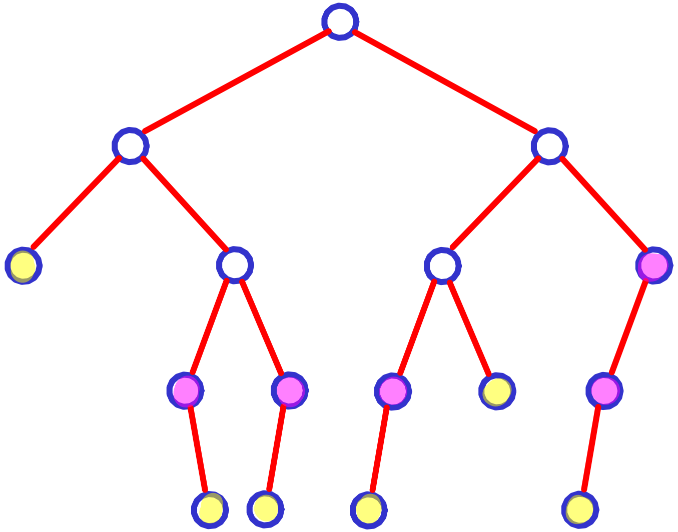
[subtrees can be empty]

leaves : 2 empty subtrees

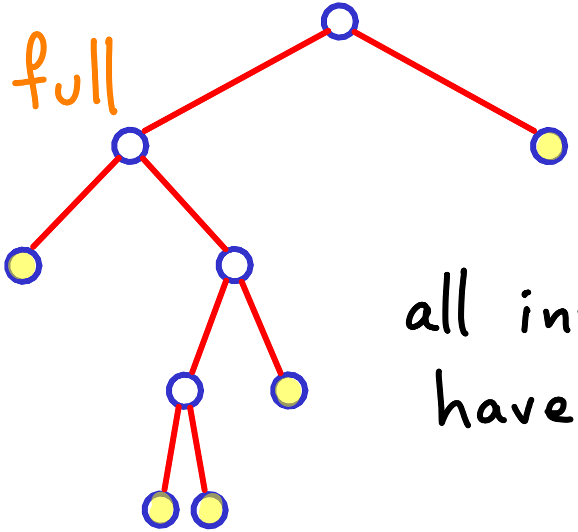
full



all internal nodes have 2 children

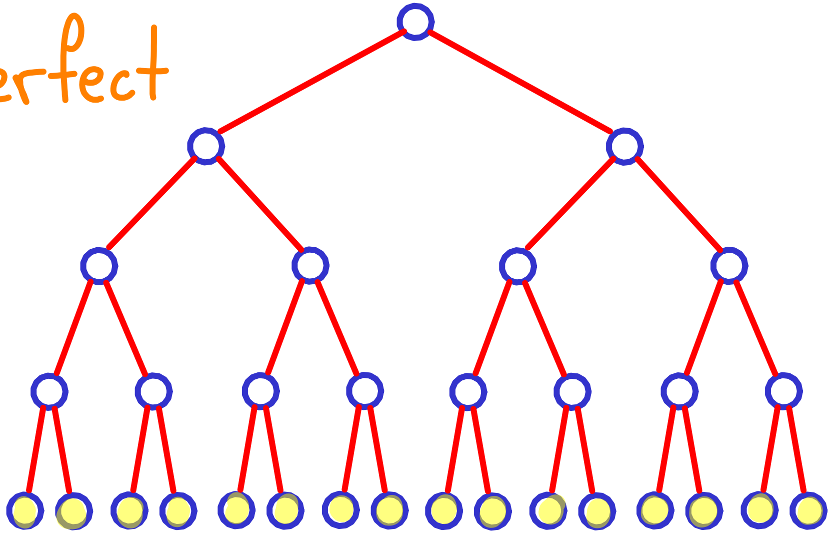


full

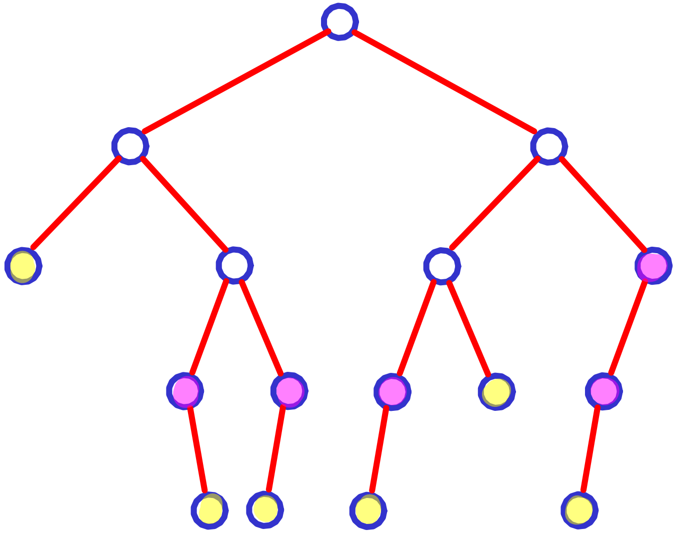


all internal nodes have 2 children

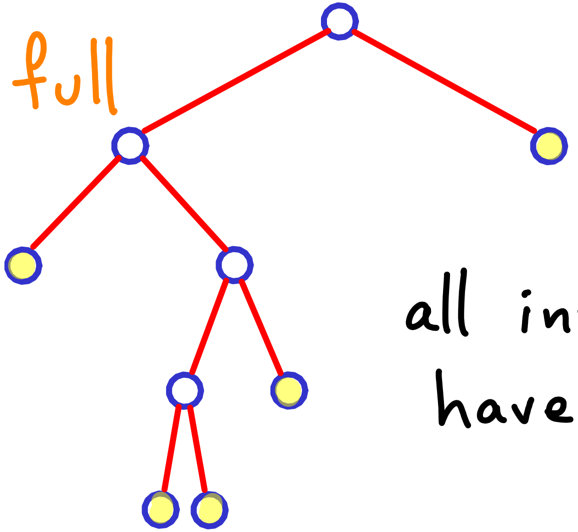
perfect



full & all leaves on same level

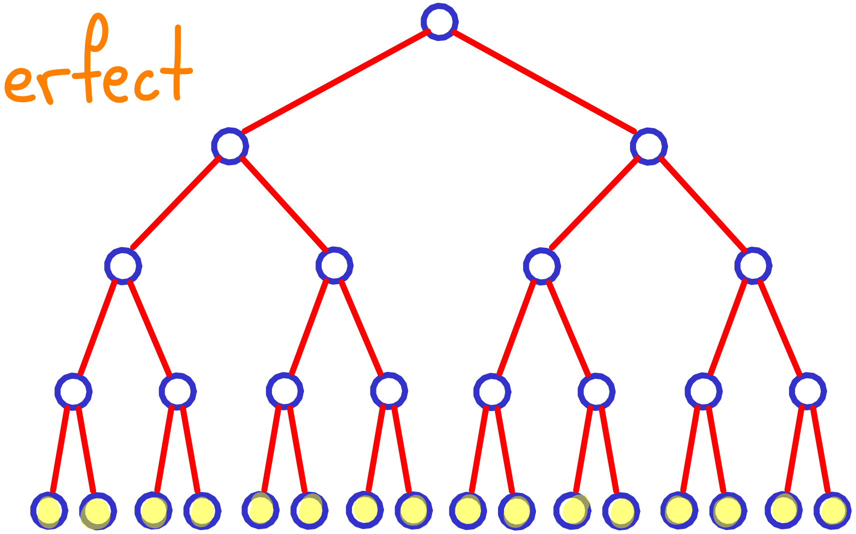


full



all internal nodes have 2 children

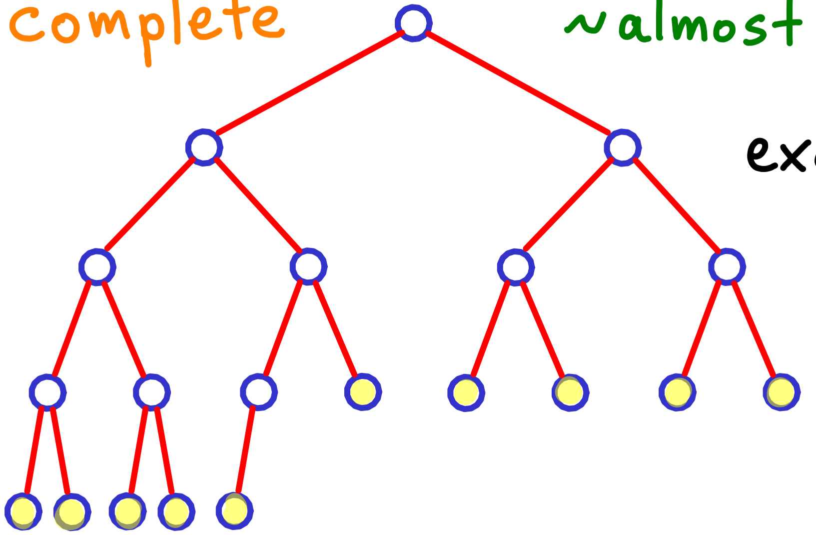
perfect



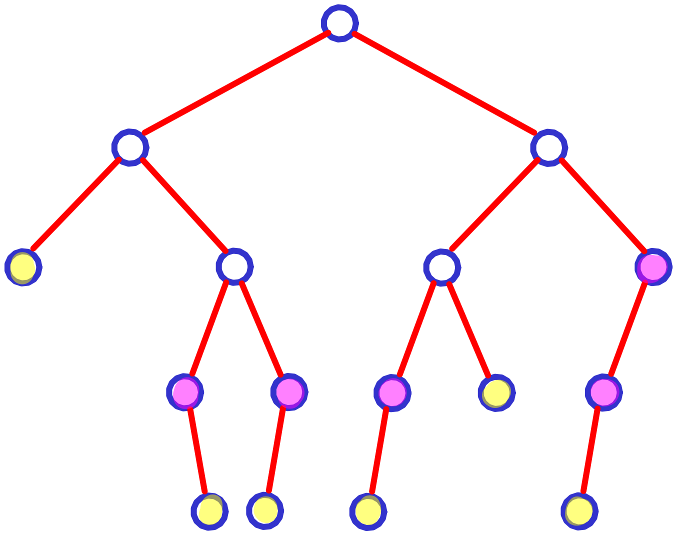
full & all leaves on same level

complete

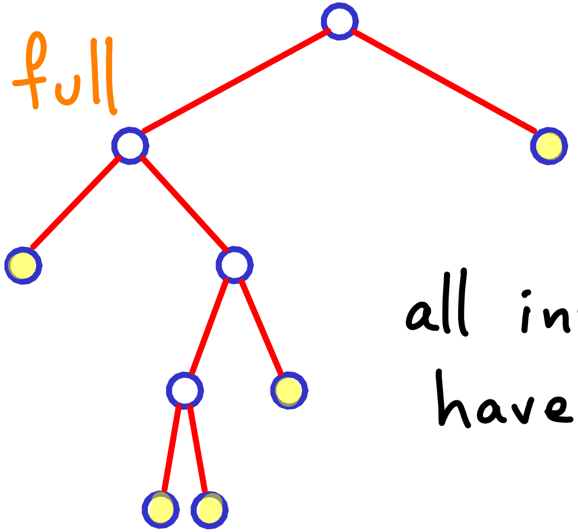
~almost perfect



exception: last level (partially) filled in from left to right

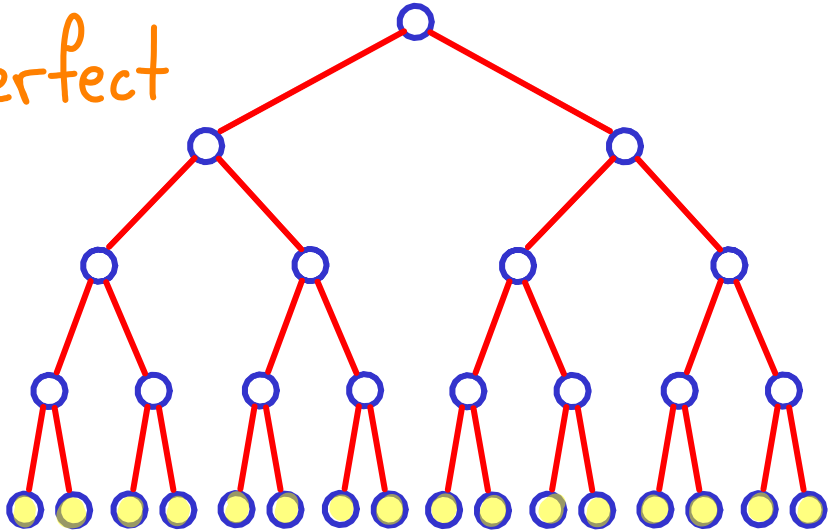


full



all internal nodes have 2 children

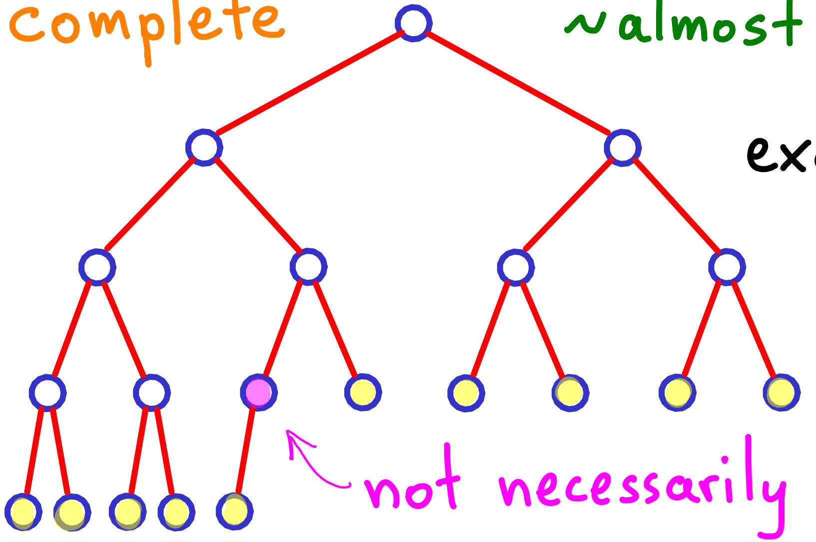
perfect



full & all leaves on same level

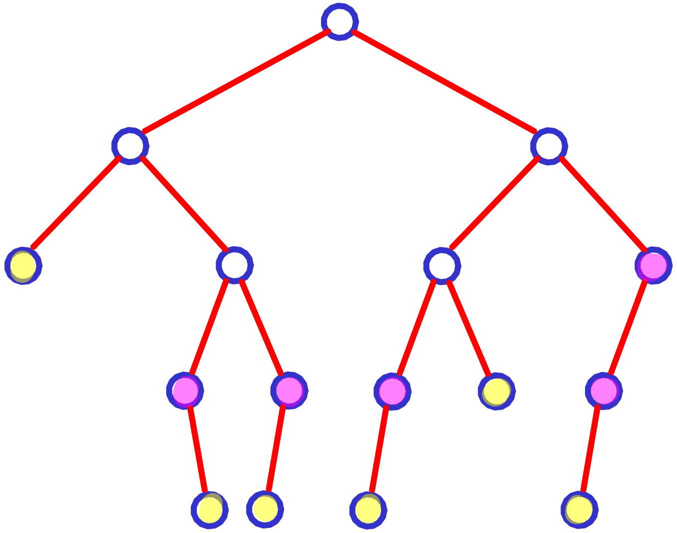
complete

~almost perfect

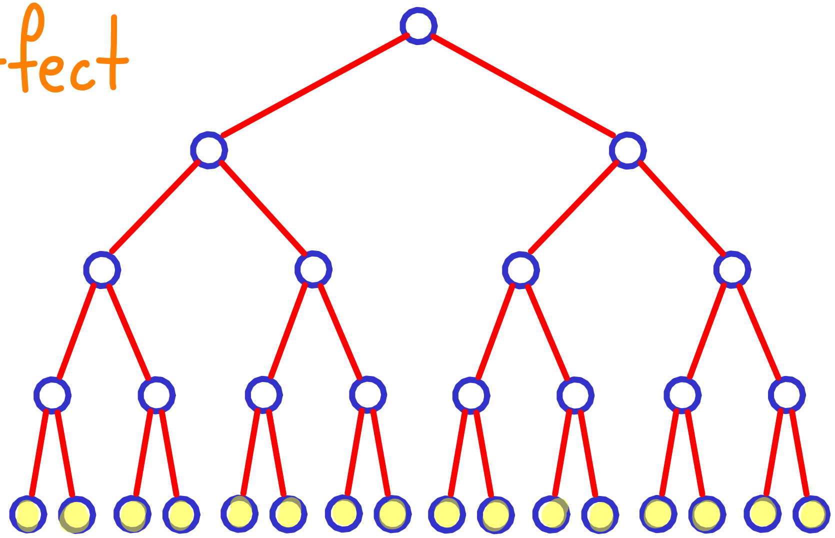


exception: last level (partially) filled in from left to right

not necessarily a full tree

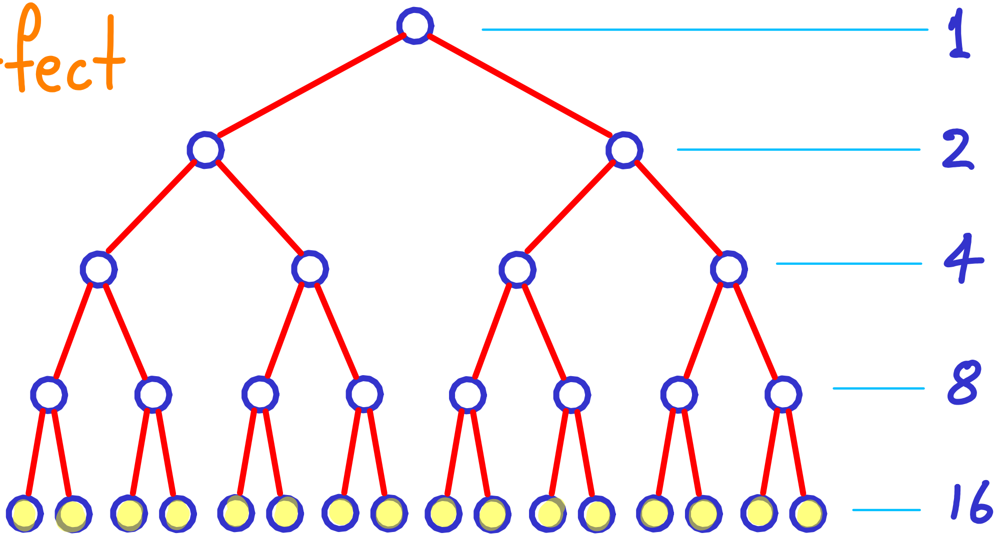


perfect



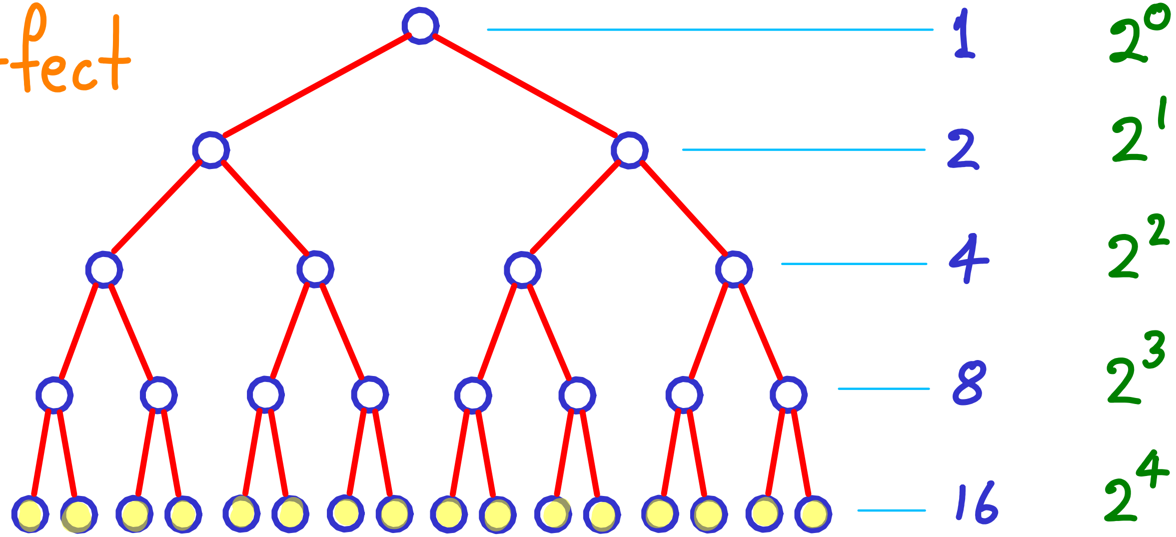
#nodes per level ?

perfect



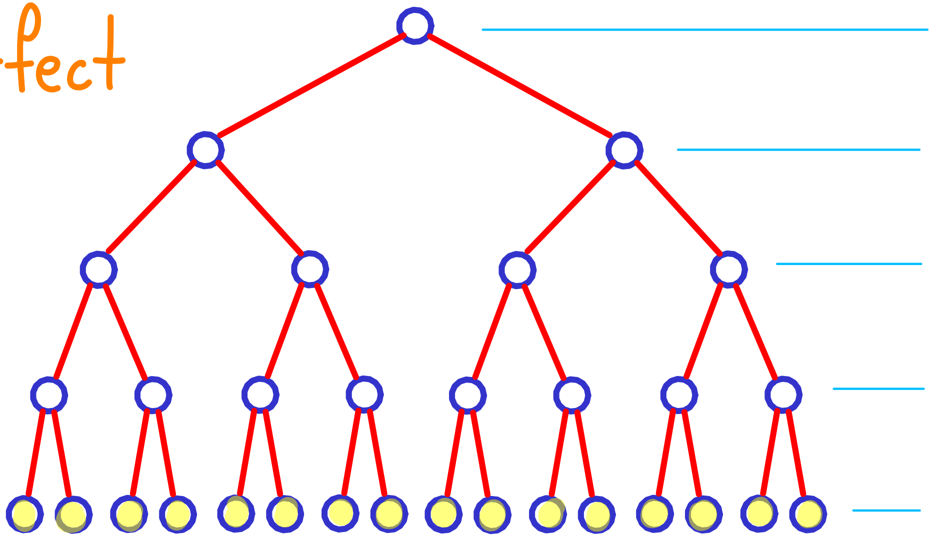
nodes per level ?

perfect



nodes per level ?

perfect

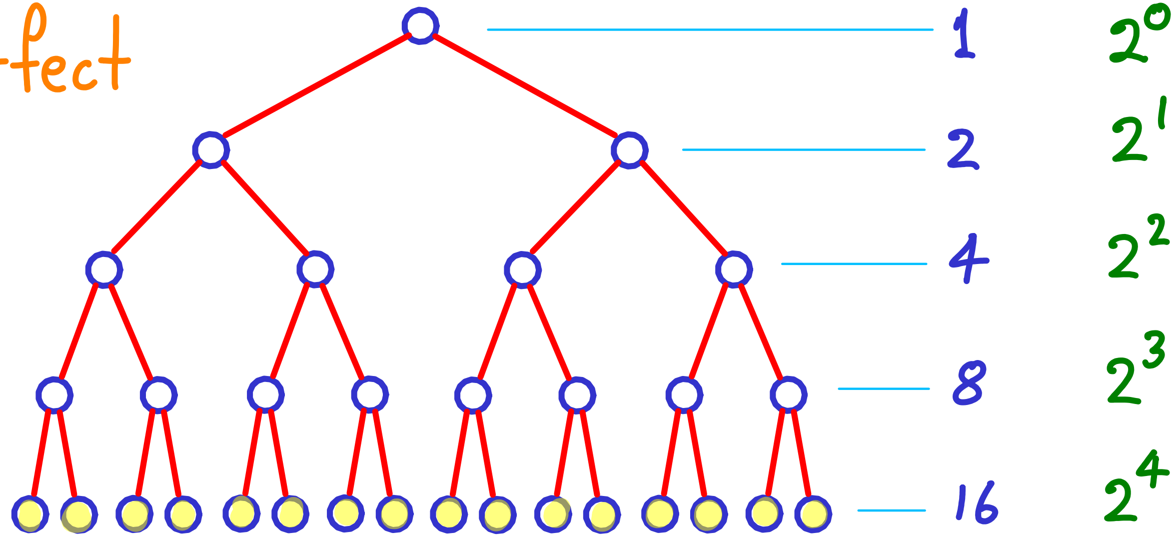


\# nodes per level ?

level i

i : 0, 1, 2, ...

perfect



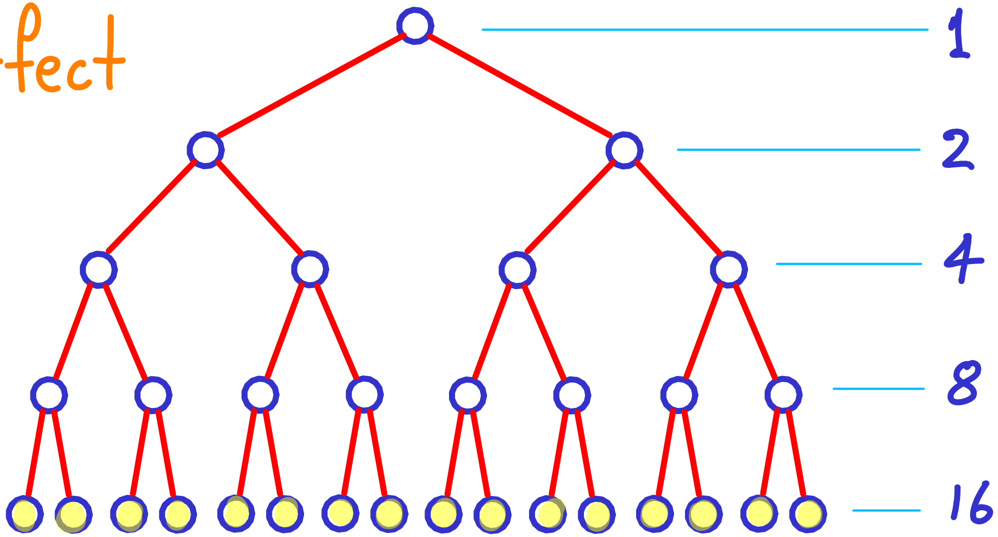
#nodes per level ?

2^i for level i

$i = \{0, 1, 2, \ldots, L-1\}$

#levels = L

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1}$ = #leaves = ℓ

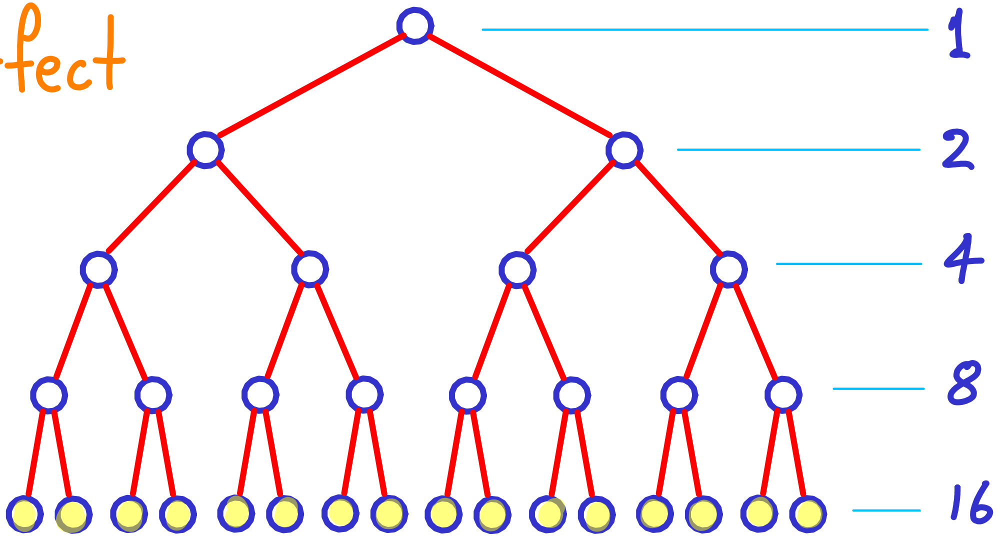
#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \dots, L-1\}$

#levels = L

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

\# nodes per level ?

$\hookrightarrow$ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

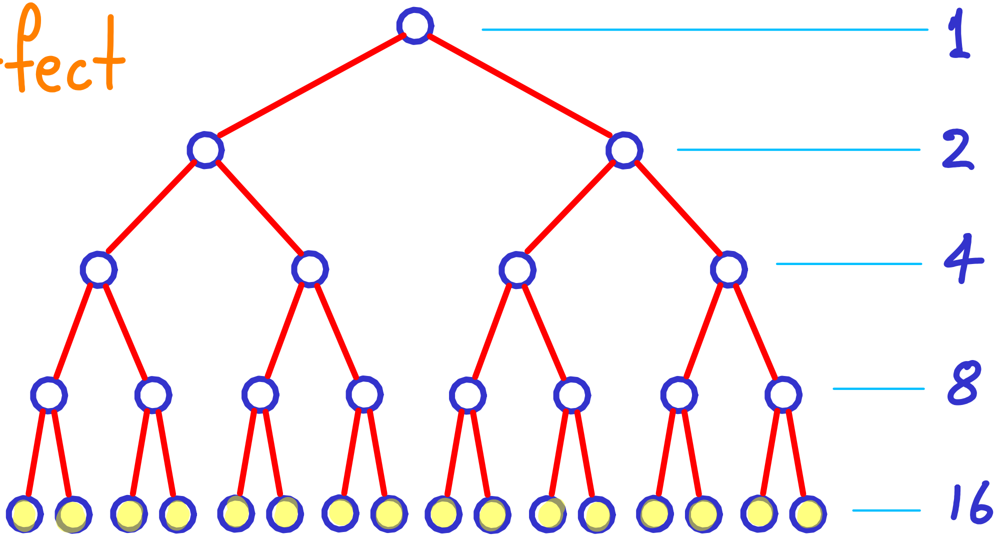
$\#\text{levels} = L$

Let n = total # nodes.

notice: n is odd

why?

perfect



2^0 : odd

2^1, 2^2, 2^3, 2^4 : all even

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

\#nodes per level ?

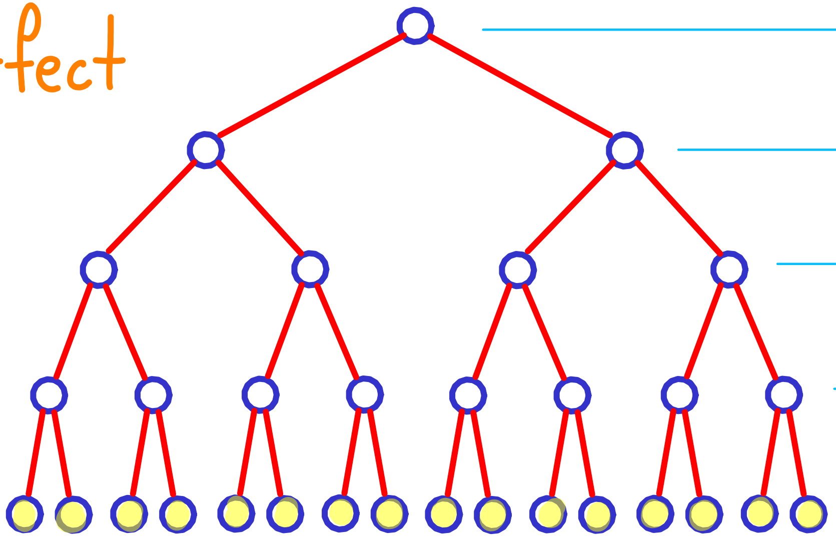
↳ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes.   notice: n is odd

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i = \{0, 1, 2, \dots, L-1\}$

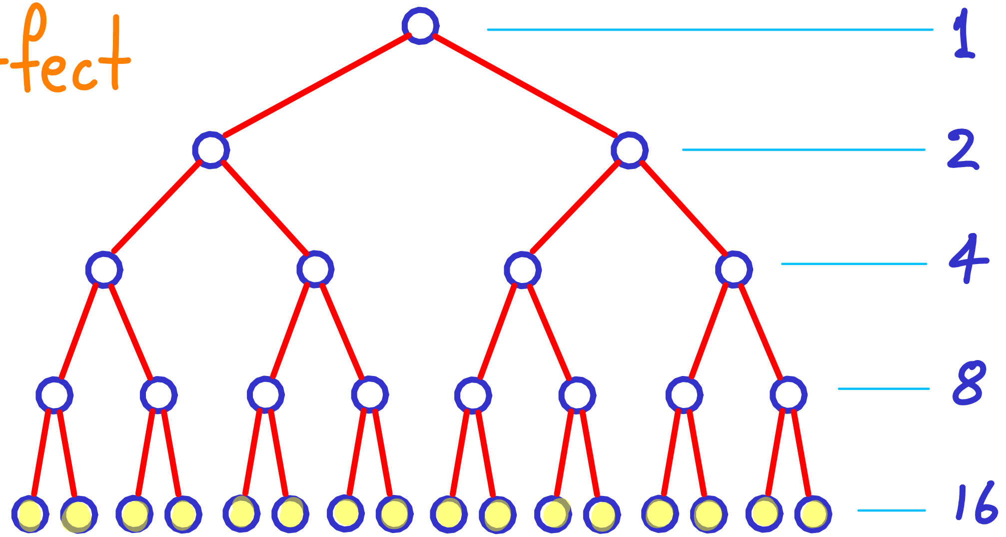
#levels = L

Let n = total #nodes. notice: n is odd

- how many levels ?

in terms of n

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i = \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

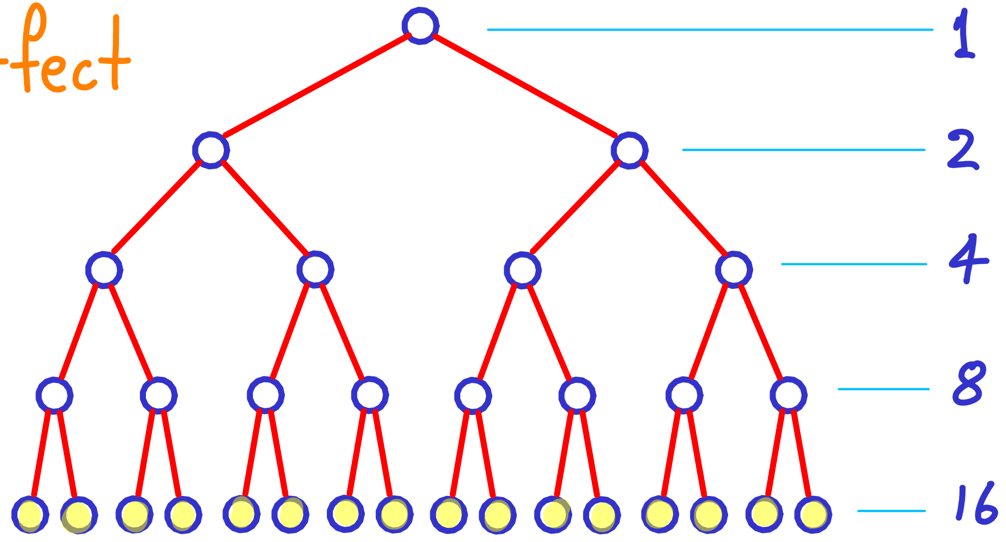
Let n = total #nodes.

notice: n is odd

- how many levels ?

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

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

\#nodes per level ?

$\hookrightarrow 2^i$ for level i

$i = \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

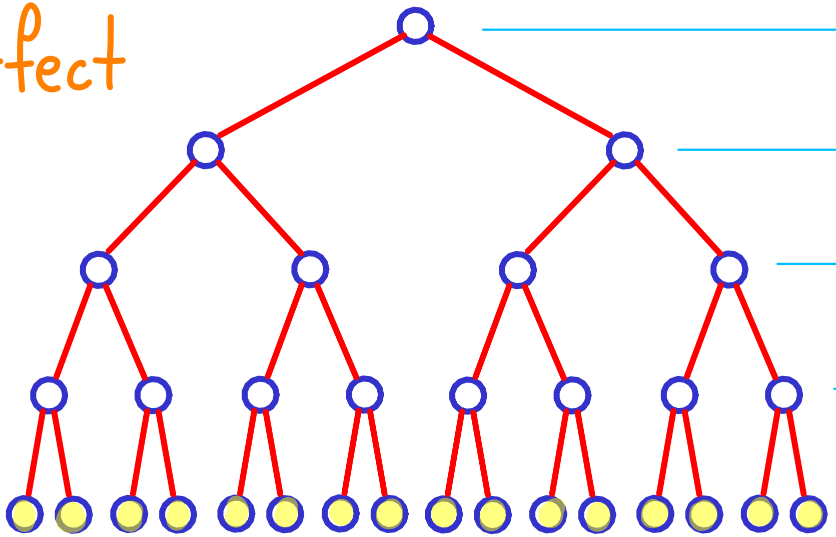
Let n = total \#nodes.

notice: n is odd

- how many levels ?

$$n = \underbrace{\sum_{i=0}^{L-1} 2^i = 2^L - 1}_{\text{classic geometric sum}}$$

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i = \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes.

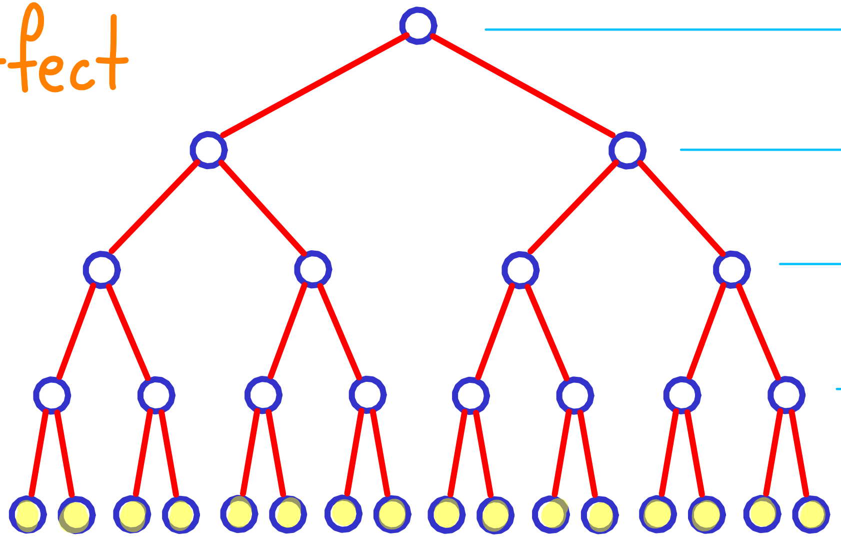
notice: n is odd

- how many levels ?

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

Base case trivial for $L=1$

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

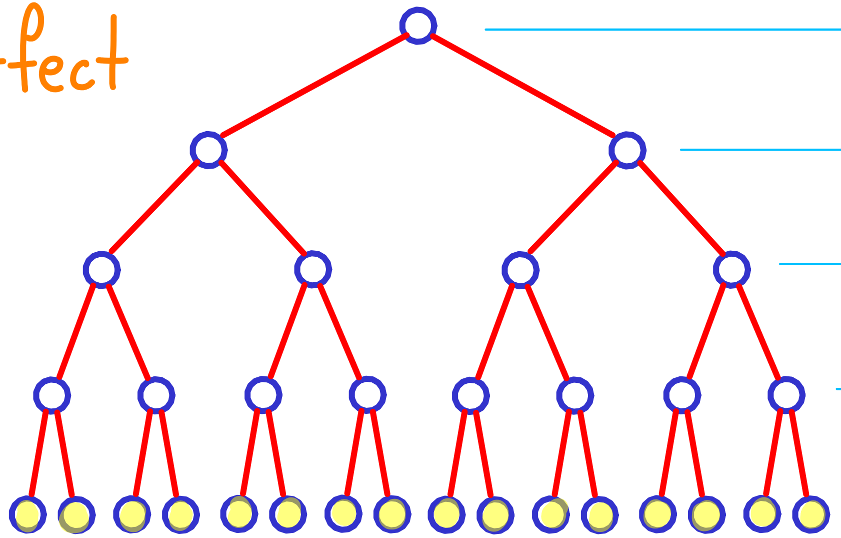
Let n = total #nodes.   notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1$

For $L \geq 2$: Hypothesis: $\sum_{i=0}^{L-2} 2^i = 2^{L-1} - 1$

Base case trivial for $L = 1$

perfect



1 $\quad 2^0$

2 $\quad 2^1$

4 $\quad 2^2$

8 $\quad 2^3$

16 $\quad 2^4 = 2^{L-1} = \#\text{leaves} = \ell$

\# nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total # nodes. notice: n is odd

- how many levels?

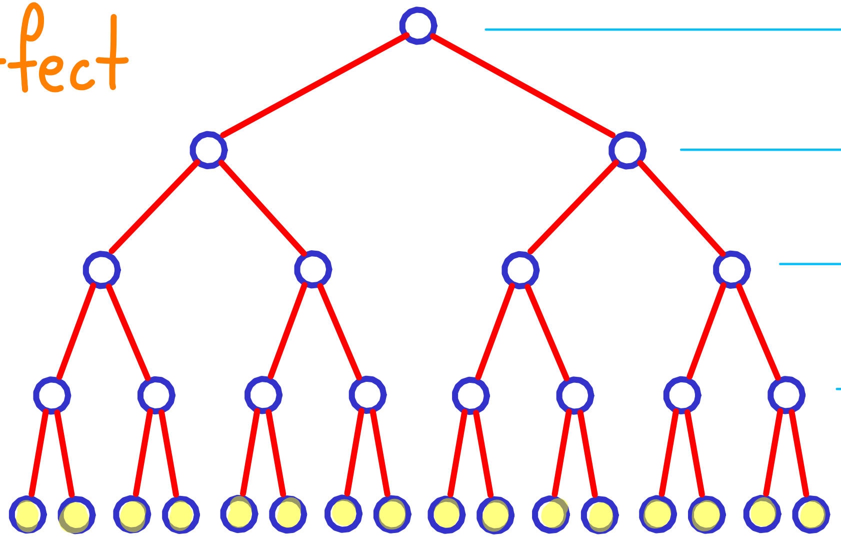
$$n = \sum_{i=0}^{L-1} 2^i = 2^L - 1$$

For $L \geq 2$: Hypothesis: $\sum_{i=0}^{L-2} 2^i = 2^{L-1} - 1$

Base case trivial for $L=1$

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

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1}$ = #leaves = ℓ

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \dots, L-1\}$

#levels = L

Let n = total #nodes. notice: n is odd

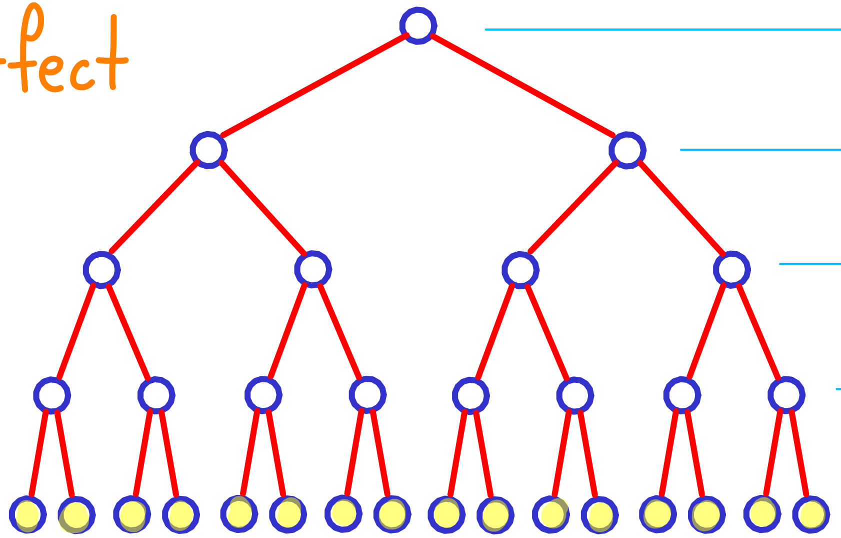
• how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1$

For $L \geq 2$: Hypothesis: $\sum_{i=0}^{L-2} 2^i = 2^{L-1} - 1$

Base case trivial for $L=1$

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

perfect



1 $\quad 2^0$

2 $\quad 2^1$

4 $\quad 2^2$

8 $\quad 2^3$

16 $\quad 2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \dots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes. notice: n is odd

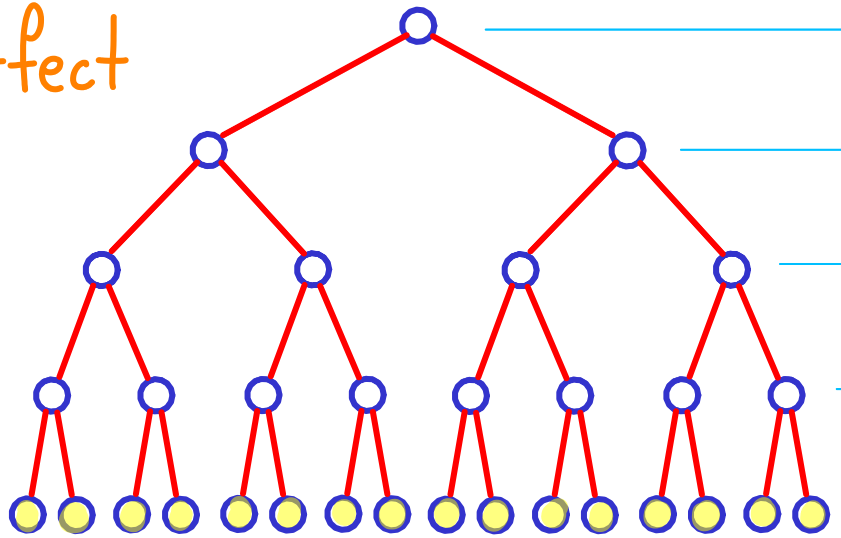
- how many levels? $\quad n = \sum_{i=0}^{L-1} 2^i = 2^L - 1$

For $L \geq 2$: Hypothesis: $\sum_{i=0}^{L-2} 2^i = 2^{L-1} - 1$

Base case trivial for $L = 1$

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

perfect



1 $\quad 2^0$

2 $\quad 2^1$

4 $\quad 2^2$

8 $\quad 2^3$

16 $\quad 2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes. notice: n is odd

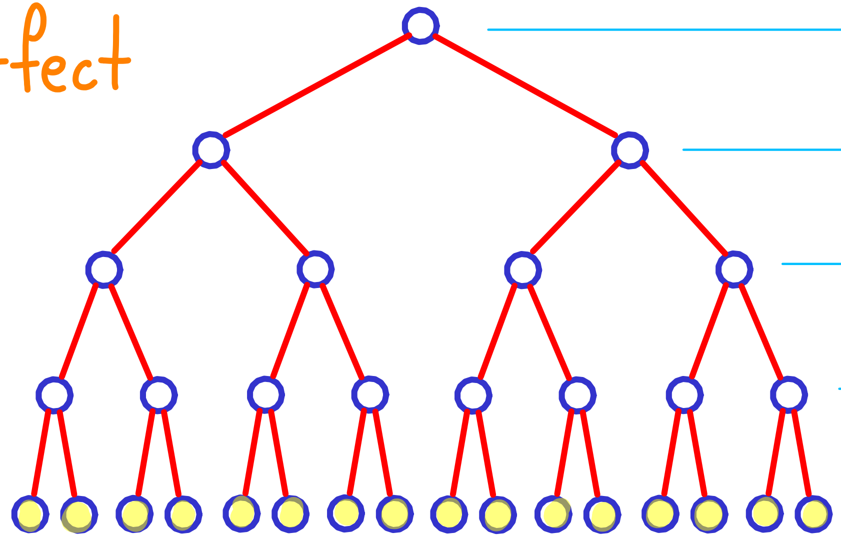
- how many levels? $\quad n = \sum_{i=0}^{L-1} 2^i = 2^L - 1$

For $L \geq 2$: Hypothesis: $\sum_{i=0}^{L-2} 2^i = 2^{L-1} - 1$

Base case trivial for $L = 1$

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

perfect



level	# nodes	
1	2^0	
2	2^1	
4	2^2	
8	2^3	
16	2^4	$= 2^{L-1}$ = #leaves $= \ell$

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

#levels $= L$

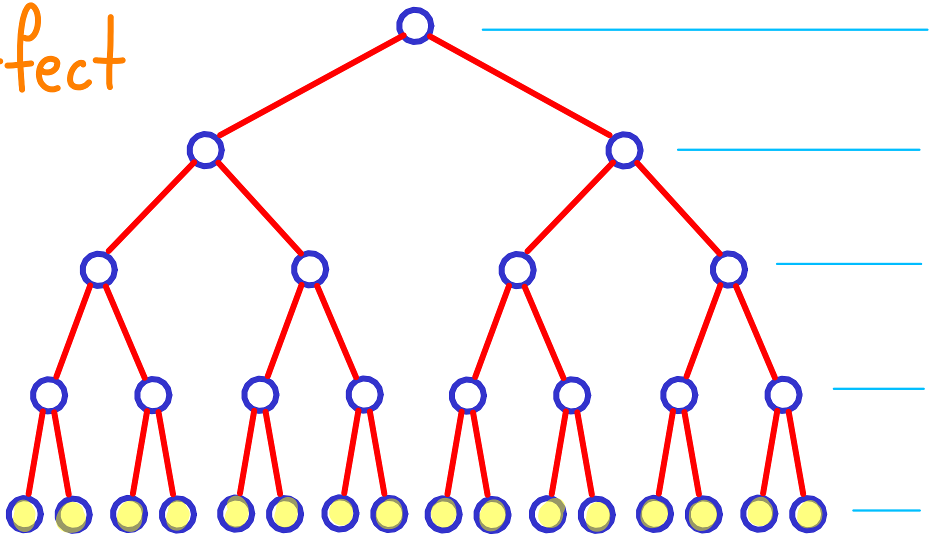
Let n = total #nodes.

notice: n is odd

- how many levels ?

$$n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \;\rightarrow\; L = \log_2(n+1)$$

perfect



1 2^0

2 2^1

4 2^2

8 2^3

16 $2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

↳ 2^i for level i

$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes.

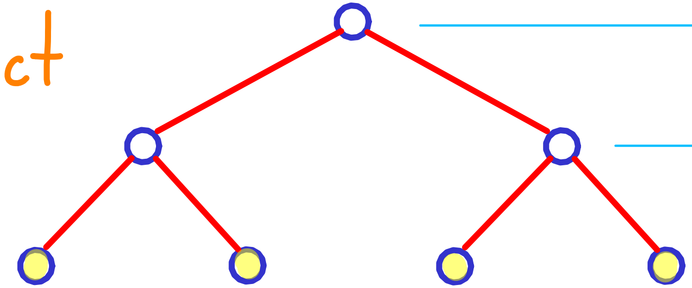
notice: n is odd

- how many levels ?

$$n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$$

- how many leaves ?

perfect



1 2^0

2 2^1

4 2^2

4 of 7

#nodes per level ?

↳ 2^i for level i

$i = \{0, 1, 2, \dots, L-1\}$

#levels = L

Let n = total #nodes.

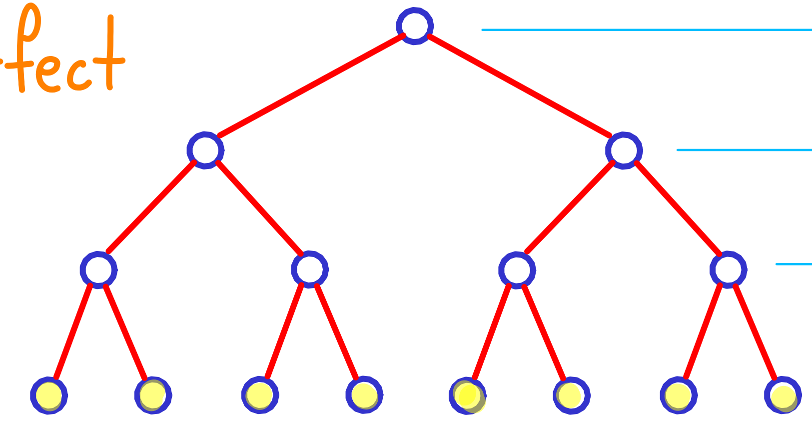
notice: n is odd

- how many levels?

$$n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$$

- how many leaves?

perfect



1 $\quad 2^0$

2 $\quad 2^1$

4 $\quad 2^2$

8 $\quad 2^3$

8 of 15

#nodes per level ?

↳ 2^i for level i

$i = \{0, 1, 2, \ldots, L-1\}$

#levels = L

Let n = total #nodes.

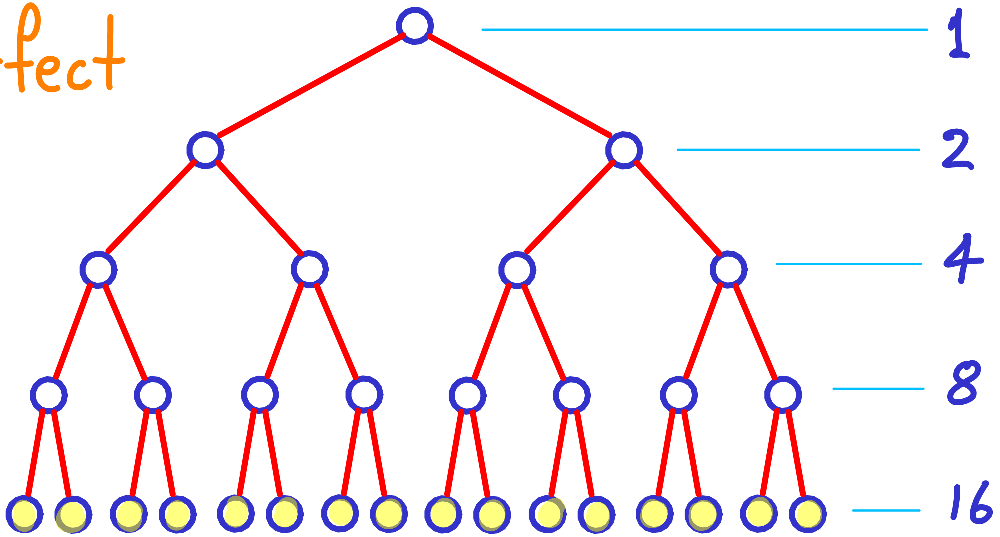
notice: n is odd

- how many levels?

$$n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \quad \rightarrow \quad L = \log_2(n+1)$$

- how many leaves?

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \ldots, L-1\}$

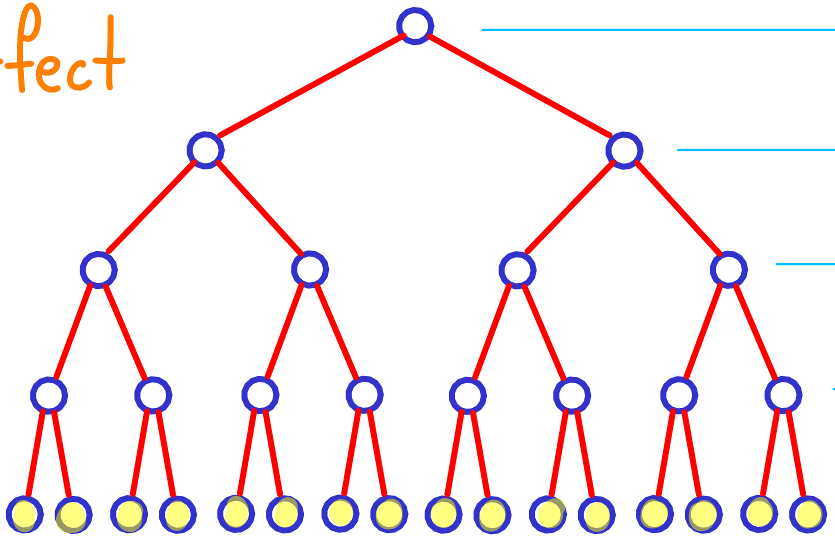
$\#\text{levels} = L$

Let n = total # nodes. notice: n is odd

- how many levels ? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$

- how many leaves ? 16 of 31

perfect



1 $\quad 2^0$

2 $\quad 2^1$

4 $\quad 2^2$

8 $\quad 2^3$

16 $\quad 2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \ldots, L-1\}$

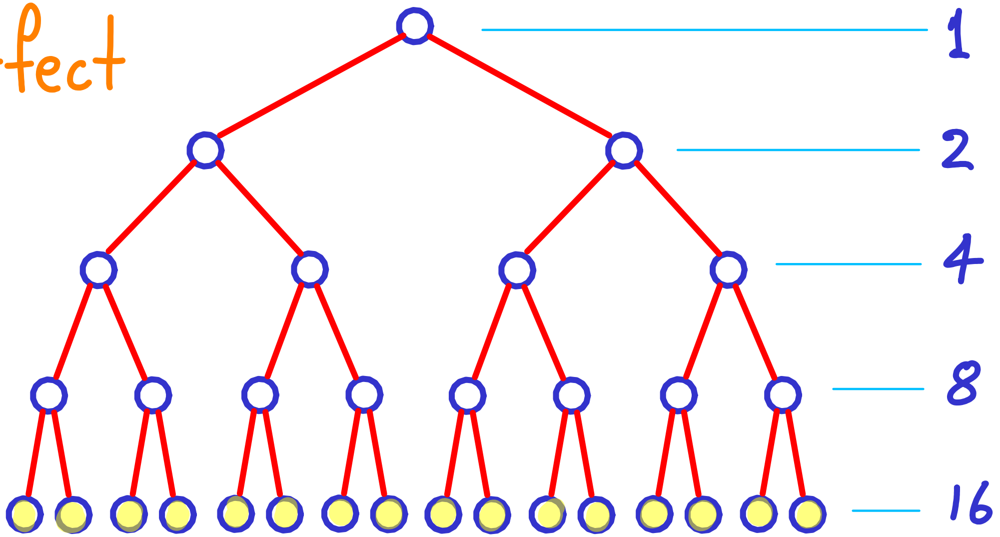
$\#\text{levels} = L$

Let n = total #nodes. notice: n is odd

- how many levels ? $\quad n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$

- how many leaves ? $\quad n = 2 \cdot 2^{L-1} - 1$

perfect



2^0

2^1

2^2

2^3

$2^4 = \underline{2^{L-1}} = \#\text{leaves} = \underline{\ell}$

\# nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \ldots, L-1\}$

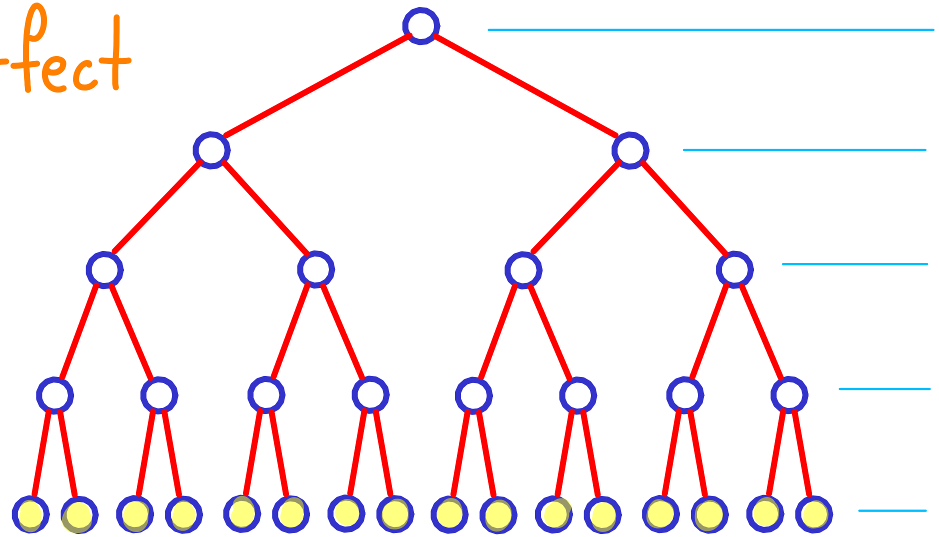
$\#\text{levels} = L$

Let n = total # nodes. notice: n is odd

- how many levels ? $\quad n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \;\rightarrow\; L = \log_2(n+1)$

- how many leaves ? $\quad n = 2 \cdot \underline{2^{L-1}} - 1 = 2\underline{\ell} - 1$

perfect



1	2^0	
2	2^1	
4	2^2	
8	2^3	
16	2^4	$= 2^{L-1}$ = #leaves = ℓ

#nodes per level ?

↳ 2^i for level i

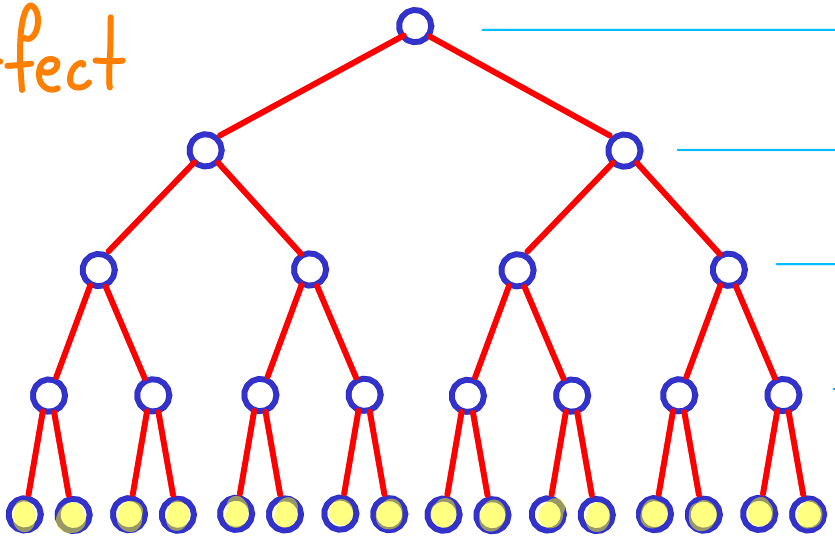
$i : \{0, 1, 2, \ldots, L-1\}$

#levels = L

Let n = total #nodes.   notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \lceil \frac{n}{2} \rceil = \frac{n+1}{2}$

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

\# nodes per level ?

↪ 2^i for level i

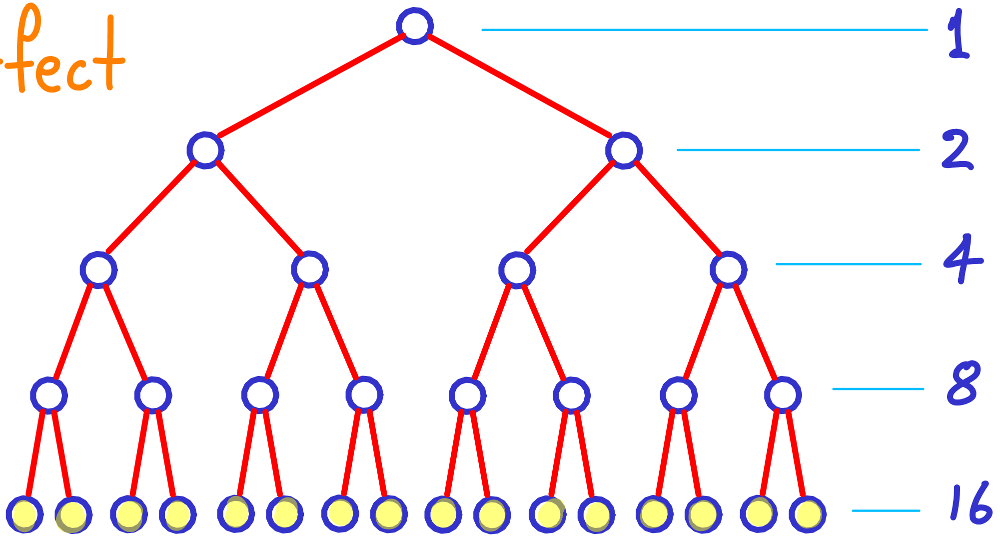
$i = \{0, 1, 2, \dots, L-1\}$

$\#\text{levels} = L$

Let n = total # nodes. notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$
- how many internal nodes?

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

$\hookrightarrow 2^i$ for level i

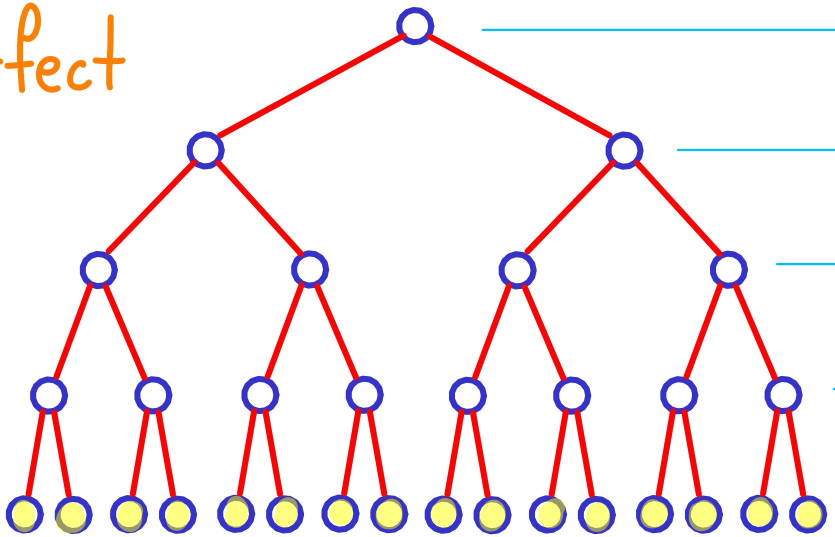
$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total #nodes. notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$
- how many internal nodes? $(n - \#\text{leaves})$

perfect



1	2^0
2	2^1
4	2^2
8	2^3
16	$2^4 = 2^{L-1}$ = #leaves = ℓ

#nodes per level ?

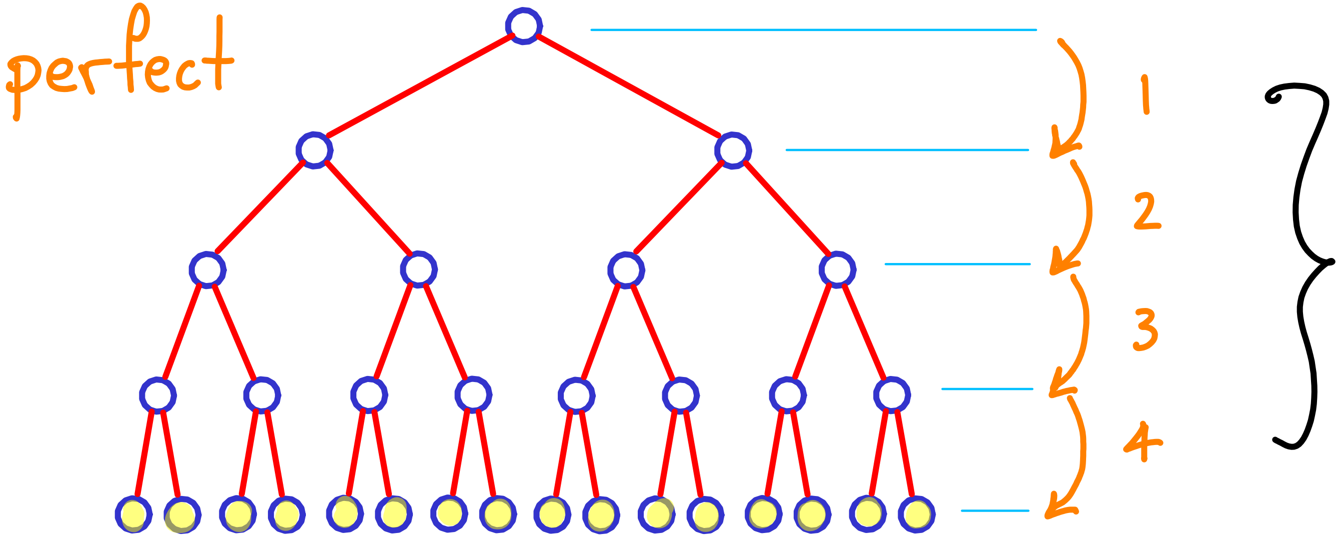
↳ 2^i for level i

$i : \{0, 1, 2, \dots, L-1\}$

#levels = L

Let n = total #nodes. notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$
- how many internal nodes? $\left\lfloor \frac{n}{2} \right\rfloor = \frac{n-1}{2}$ (n − #leaves)

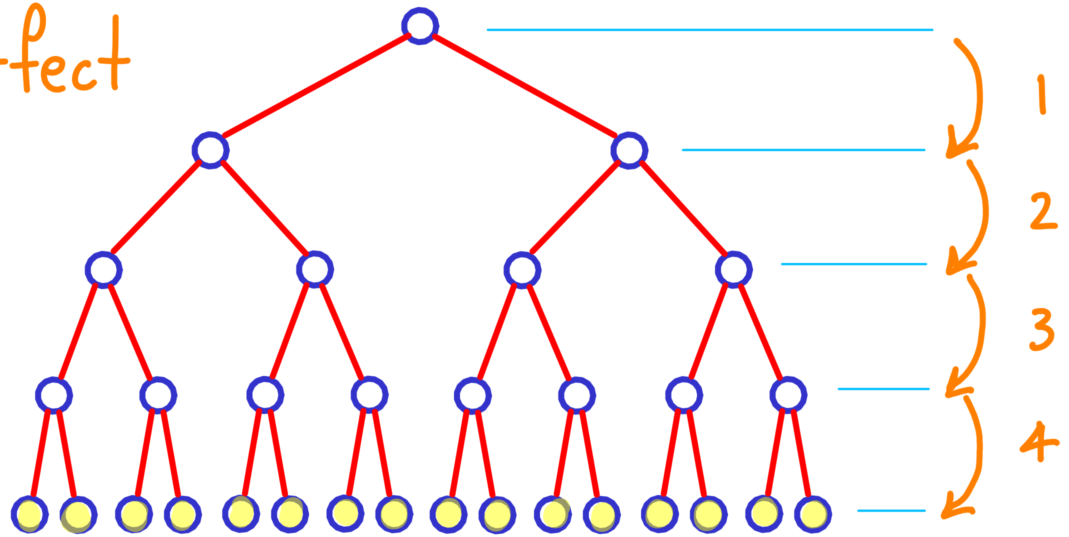


depth of tree

$\Updownarrow$

height of tree

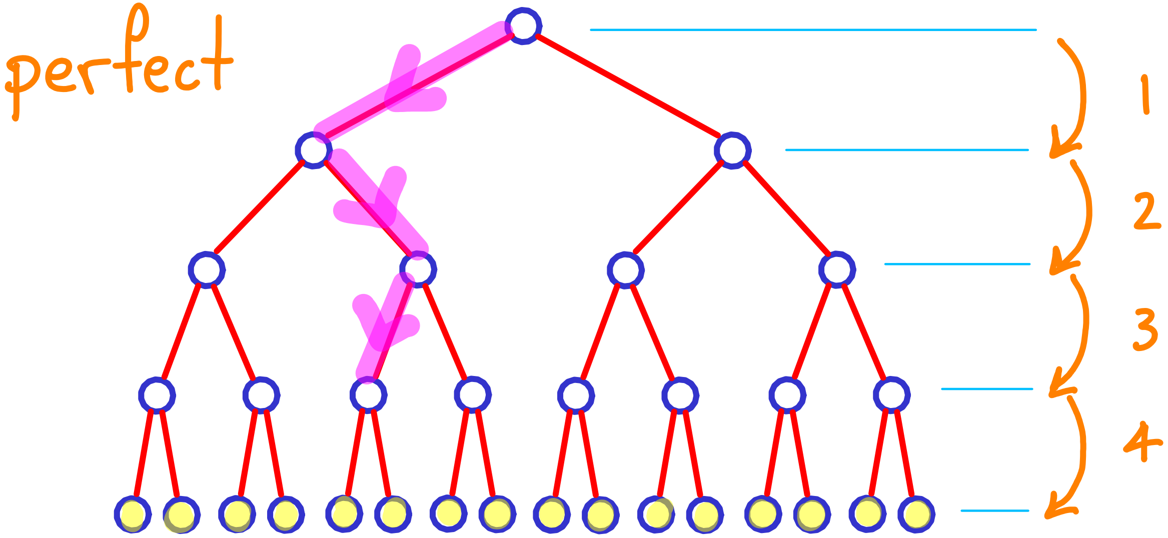
perfect



depth of tree $\Updownarrow$ height of tree : h

h = number of "steps" (edges) from root to deepest leaf

lowest level



depth of tree

$\Updownarrow$

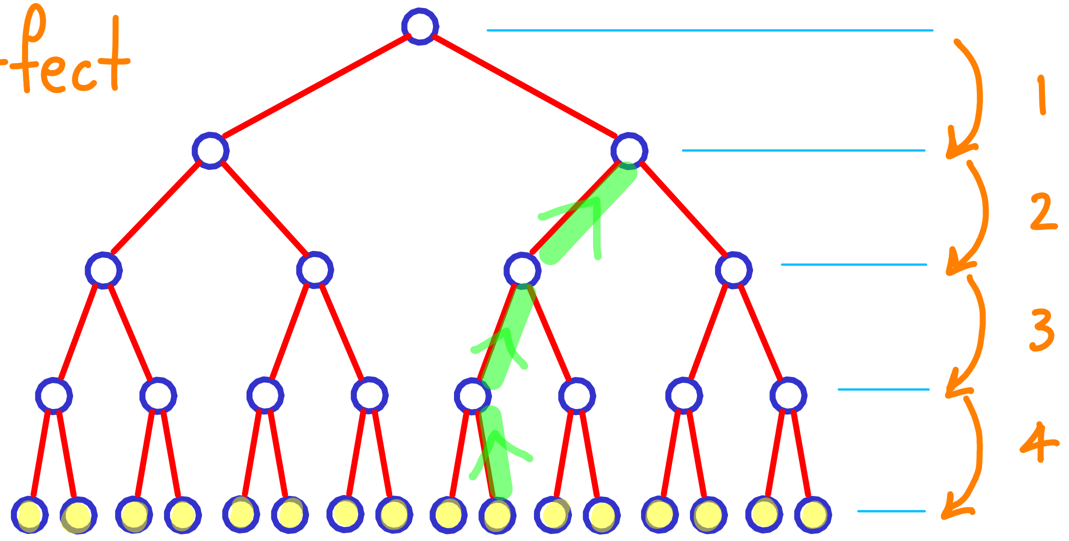
height of tree : h

h = number of "steps" (edges) from root to deepest leaf

lowest level

- depth of node = # steps from root

perfect



depth of tree

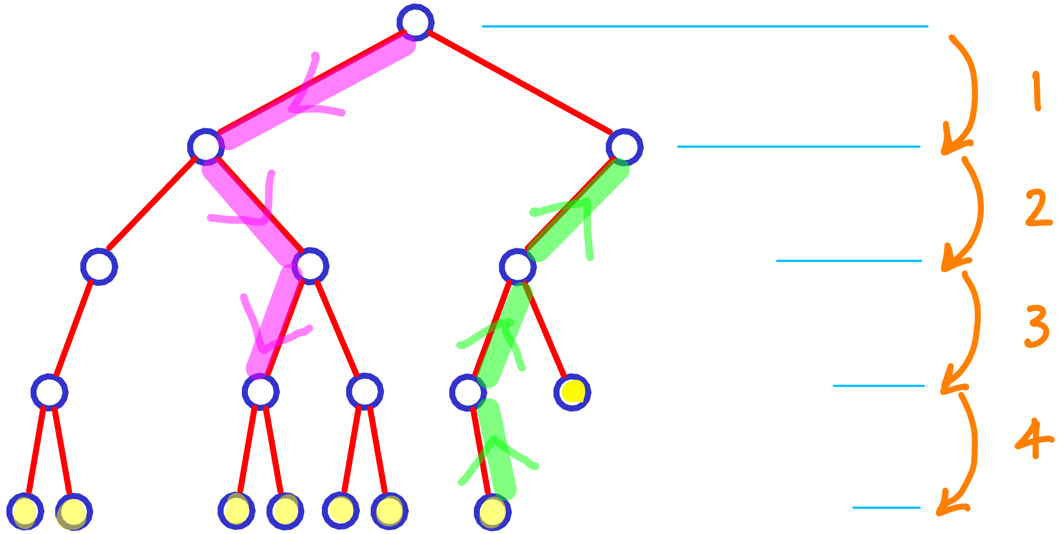
$\Updownarrow$

height of tree : h

h = number of "steps" (edges) from root to deepest leaf

lowest level

- depth of node = #steps from root
- height of node = #steps from deepest descendant

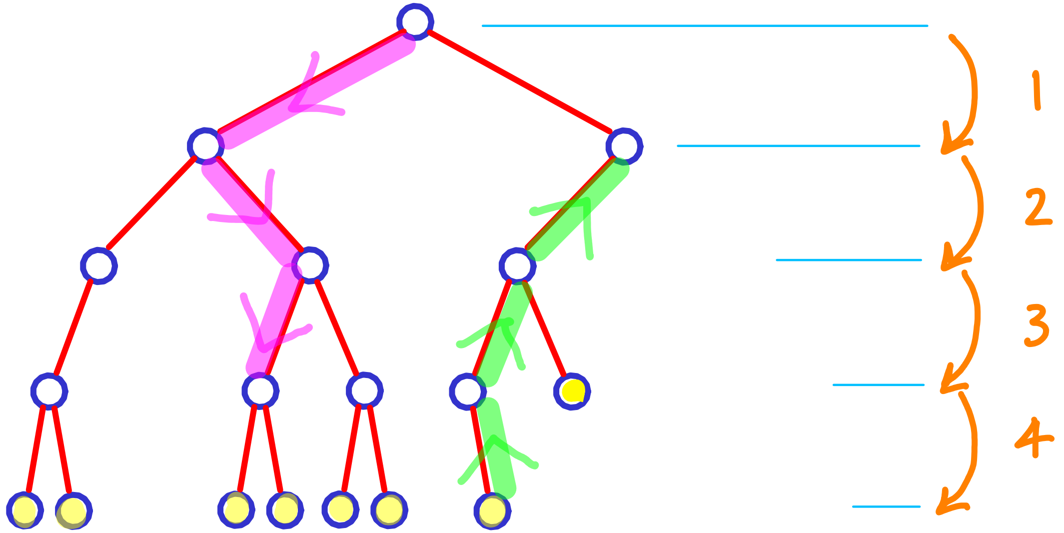


depth of tree $\Updownarrow$ height of tree : h

h = number of "steps" (edges) from root to deepest leaf

lowest level

- depth of node = #steps from root
- height of node = #steps from deepest descendant
- definitions hold for any tree



depth of tree

⇕

height of tree : h

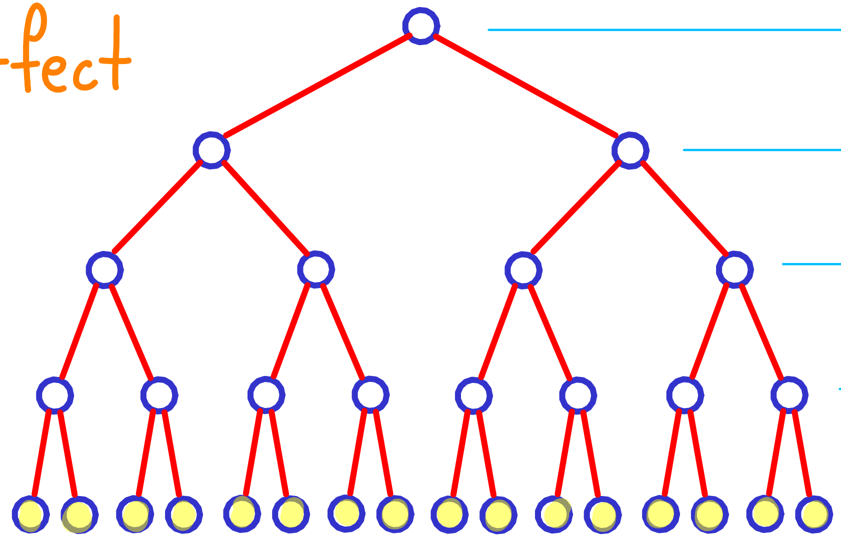
h = number of "steps" (edges) from root to deepest leaf

lowest level

- depth of node = #steps from root
- height of node = #steps from deepest descendant
- definitions hold for any tree

$\boxed{h = L - 1}$

perfect



1	2^0	
2	2^1	
4	2^2	
8	2^3	
16	2^4	$= 2^{L-1} = \#\text{leaves} = \ell$

\# nodes per level ?

↳ 2^i for level i

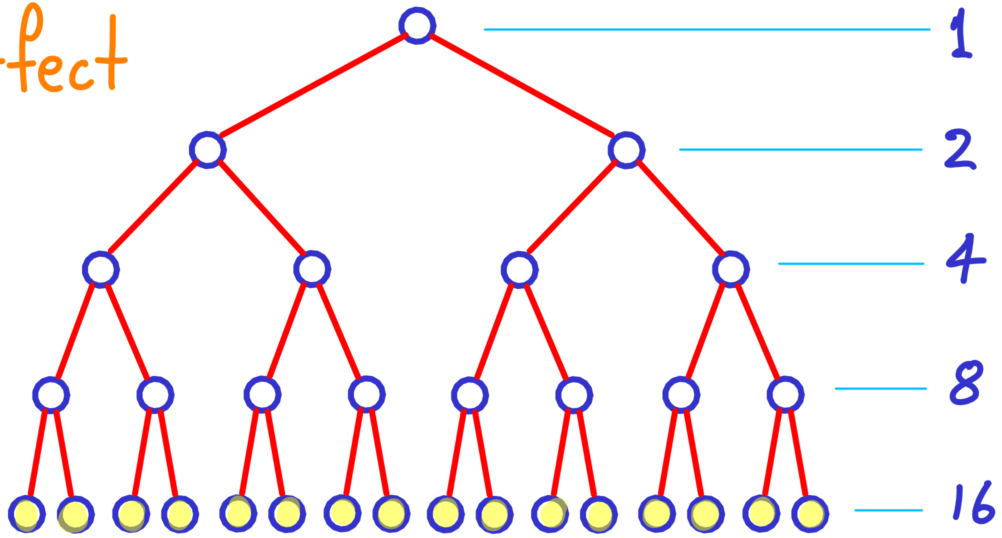
$i : \{0, 1, 2, \ldots, L-1\}$

$\#\text{levels} = L$

Let n = total # nodes. notice: n is odd

- how many levels? $\quad n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $\quad n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$
- how many internal nodes? $\quad \left\lfloor \frac{n}{2} \right\rfloor = \frac{n-1}{2} \quad (n - \#\text{leaves})$

perfect



2^0

2^1

2^2

2^3

$2^4 = 2^{L-1} = \#\text{leaves} = \ell$

#nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \dots, L-1\}$

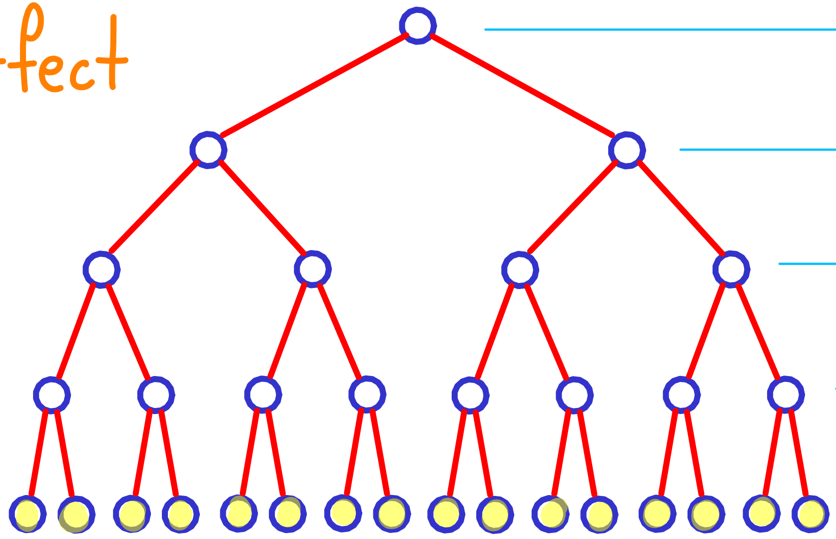
$\#\text{levels} = L$

$\boxed{h = L-1}$

Let n = total #nodes.   notice: n is odd

- how many levels? $n = \sum_{i=0}^{L-1} 2^i = 2^L - 1 \rightarrow L = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \lceil \frac{n}{2} \rceil = \frac{n+1}{2}$
- how many internal nodes? $\lfloor \frac{n}{2} \rfloor = \frac{n-1}{2}$ $(n - \#\text{leaves})$

perfect



1 2^0

2 2^1

4 2^2

8 2^3

16 $2^4 = 2^h$ = #leaves = ℓ

#nodes per level ?

$\hookrightarrow 2^i$ for level i

$i : \{0, 1, 2, \dots, L-1\}$

#levels = L

$\boxed{h = L-1}$

Let n = total #nodes. notice: n is odd

- how many levels? $n = \sum_{i=0}^{h} 2^i = 2^{h+1} - 1 \rightarrow h+1 = \log_2(n+1)$
- how many leaves? $n = 2 \cdot 2^{L-1} - 1 = 2\ell - 1 \rightarrow \ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$
- how many internal nodes? $\left\lfloor \frac{n}{2} \right\rfloor = \frac{n-1}{2}$ (n − #leaves)

summary

ℓ vs h

$$\ell = 2^h$$

$$h = \log_2 \ell$$

ℓ vs n

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

$$\ell = \#\text{internal nodes} + 1$$

h vs n

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

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

	Exact summary	Approximate summary
ℓ vs h	$\ell = 2^h$ $h = \log_2 \ell$	nothing to approximate
ℓ vs n	$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$ $\ell = \#\text{internal nodes} + 1$	
h vs n	$h+1 = \log_2(n+1)$ $n = 2^{h+1} - 1$	

	Exact summary	Approximate summary
ℓ vs h	$\ell = 2^h$	$\leftarrow$
	$h = \log_2 \ell$	$\leftarrow$
ℓ vs n	$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$	within ± 1 $\ell \approx \frac{n}{2}$
	$\ell = \#\text{internal nodes} + 1$	$\ell \approx \#\text{internal nodes}$
h vs n	$h+1 = \log_2(n+1)$	
	$n = 2^{h+1} - 1$	

Exact summary

ℓ vs h

$$\ell = 2^h$$

$$h = \log_2 \ell$$

ℓ vs n

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

$$\ell = \#\text{internal nodes} + 1$$

h vs n

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

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

Approximate summary

$\leftarrow$

$\leftarrow$

within ±1

$$\ell \approx \frac{n}{2}$$

$$\ell \approx \#\text{internal nodes}$$

$$h \approx \log_2 n$$

$h = \Theta(\log n)$

	Exact summary	Approximate summary
ℓ vs h	$\ell = 2^h$	$\leftarrow$
	$h = \log_2 \ell$	$\leftarrow$
		within ± 1
ℓ vs n	$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$	$\ell \approx \frac{n}{2}$
	$\ell = \#\text{internal nodes} + 1$	$\ell \approx \#\text{internal nodes}$
h vs n	$h+1 = \log_2(n+1)$	$h \approx \log_2 n$ — $h = \Theta(\log n)$
		within factor of 2
	$n = 2^{h+1} - 1$	$n \approx 2^h$ — $n = \Theta(2^h)$

For perfect binary trees, we saw:

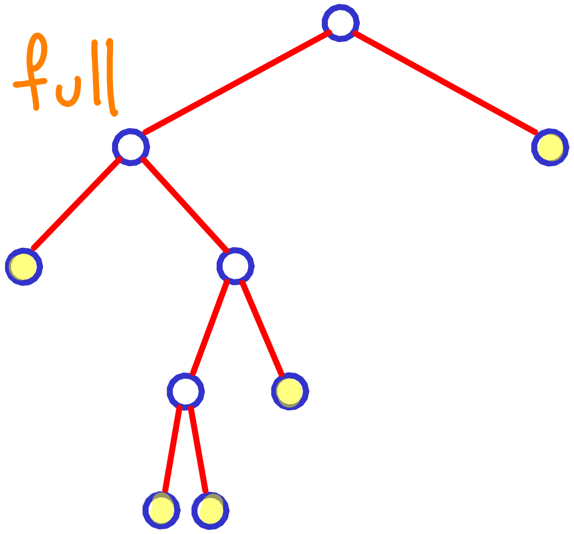
ℓ vs n

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

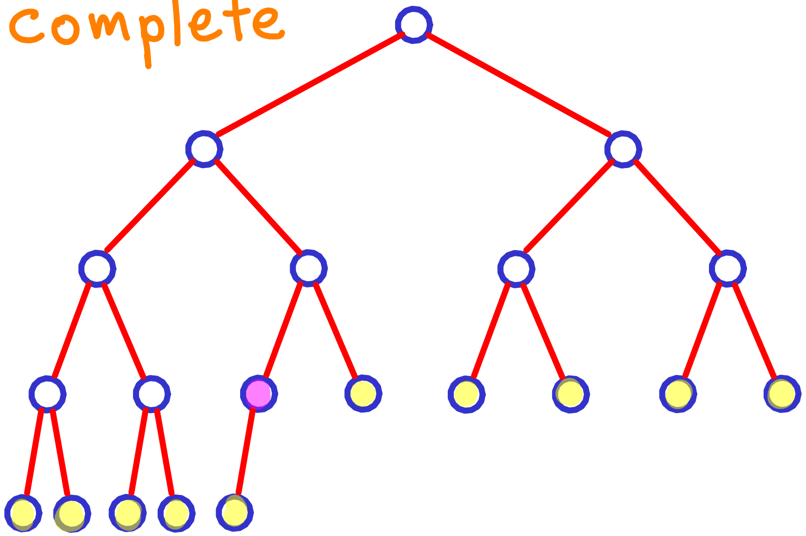
$$\ell = \#\text{internal nodes} + 1$$

What about:

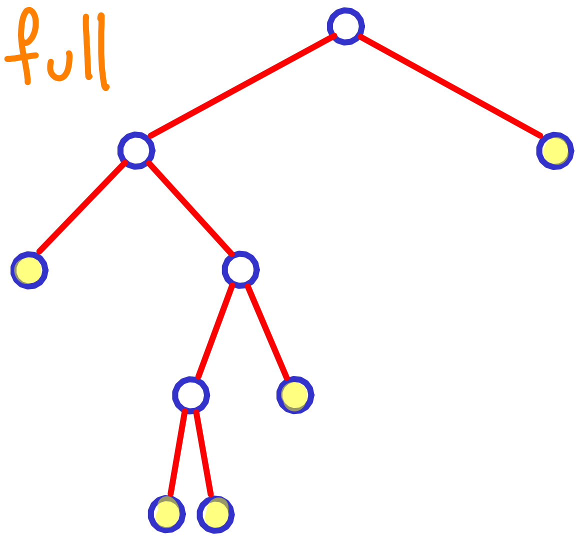
full



complete



full



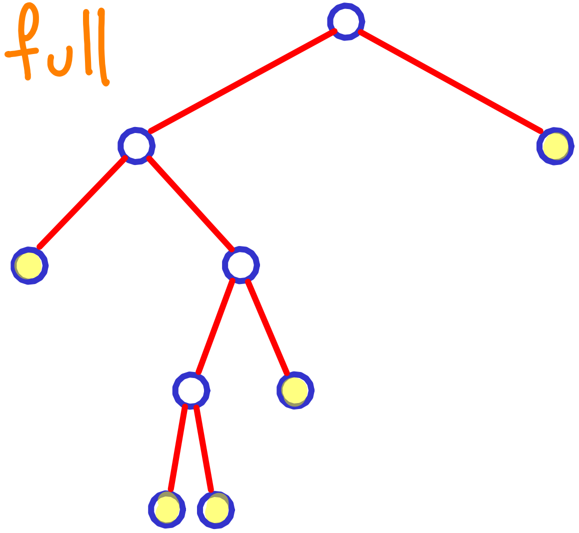
ℓ = # internal nodes + 1

?

5 leaves

4 internal

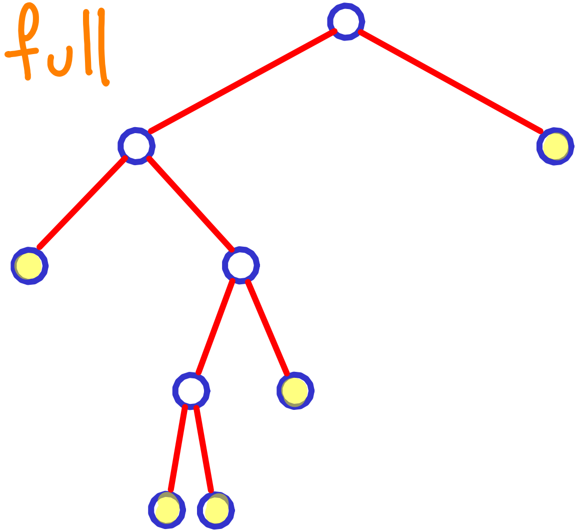
full



$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

Induction on i

full



ℓ = # internal nodes + 1 = $i+1$

Induction on i

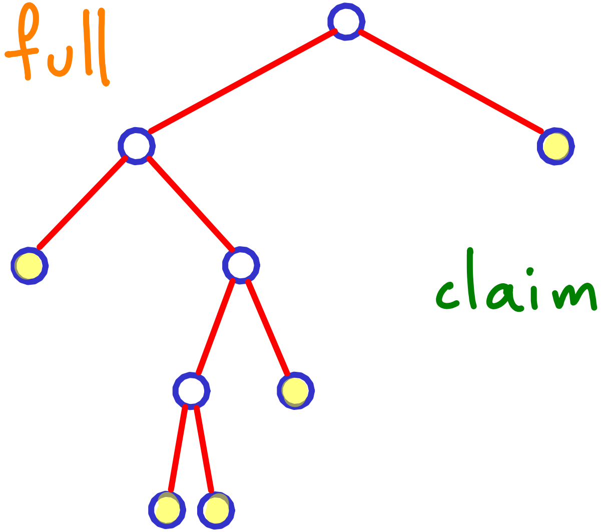
base case

trivial

$\ell=1 \quad i=0$

$n=1$

full



ℓ = # internal nodes + 1 = $i+1$

Induction on i

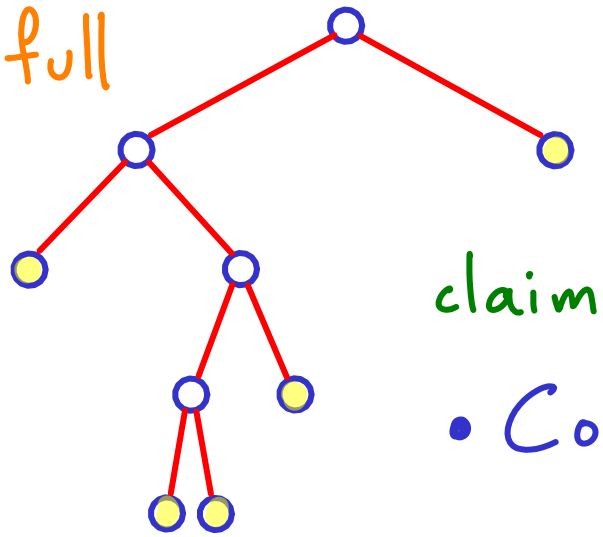
To prove for $i \geqslant 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

base case
trivial

$\ell = 1 \quad i = 0$

$n = 1$

full



ℓ = #internal nodes + 1 = $i+1$

Induction on i

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

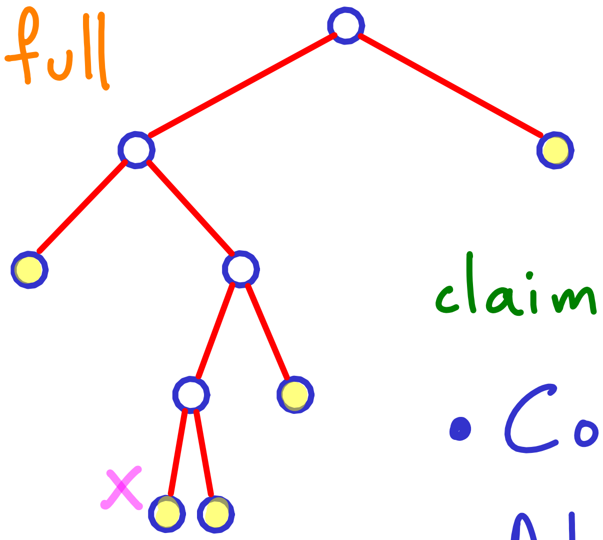
- Consider any full tree with i internal nodes.

base case trivial

$\ell = 1 \quad i = 0$

$n = 1$

full



$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

Induction on i

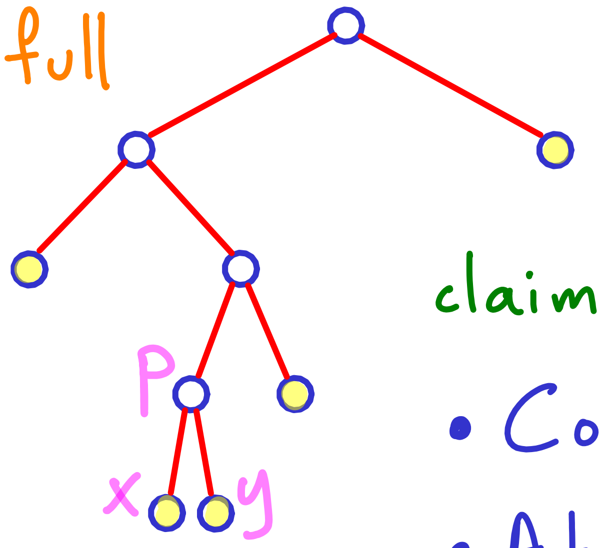
To prove for $i \geqslant 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.

base case trivial

$\ell = 1 \quad i = 0$

$n = 1$



ℓ = # internal nodes + 1 = $i+1$

Induction on i

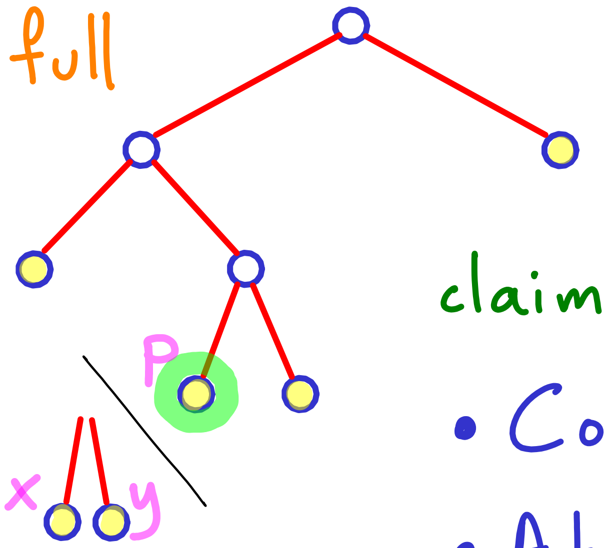
To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)

base case
trivial

$\ell = 1 \quad i = 0$

$n = 1$



base case trivial

$\ell=1 \quad i=0$

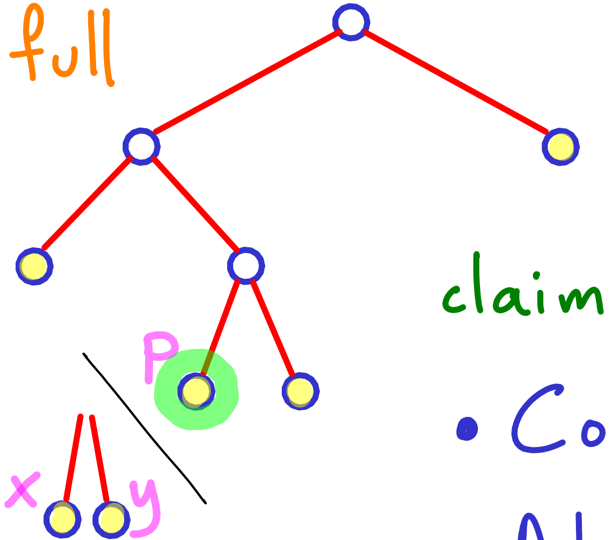
$n=1$

$\ell = \#\text{internal nodes} + 1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.



ℓ = #internal nodes + 1 = $i+1$

Induction on i

base case trivial

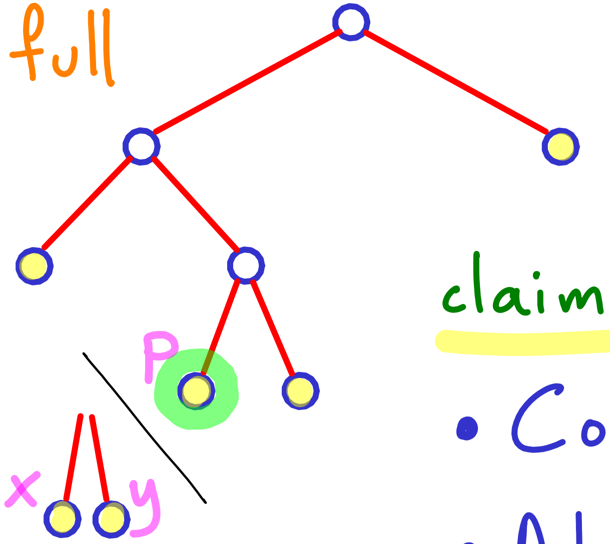
$\boxed{\ell=1 \quad i=0 \quad n=1}$

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.
- New tree has $i-1$ internal nodes & is full.

$\ell = \underline{\#\text{internal nodes}} + 1 = i+1$

Induction on i



base case trivial

$\ell = 1 \quad i = 0$

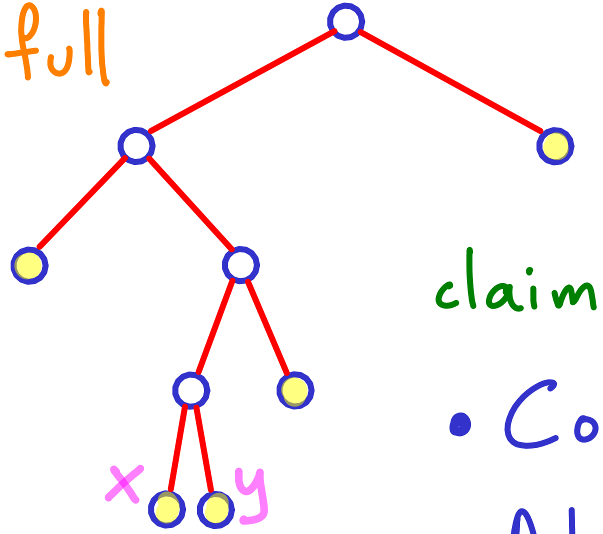
$n = 1$

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.
- New tree has $i-1$ internal nodes & is full.

↳ has i leaves by hypothesis.

full



$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

Induction on i

base case trivial

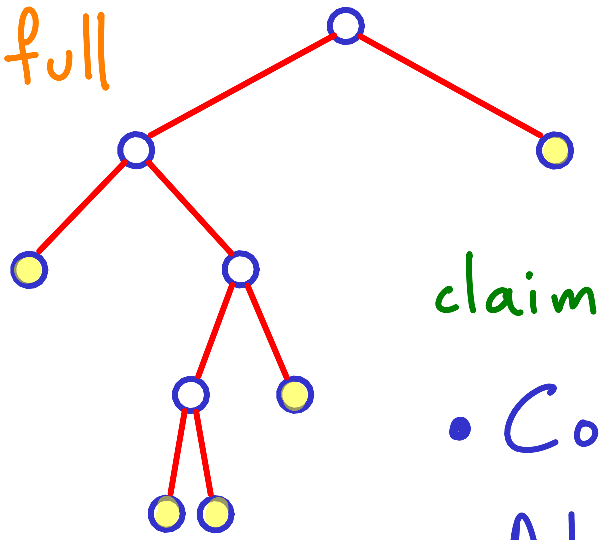
$\ell = 1 \quad i = 0$

$n = 1$

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.
- New tree has $i-1$ internal nodes & is full.
  ↳ has i leaves by hypothesis.
- Put x & y back. +2 leaves

full



base case trivial

$$\boxed{\begin{array}{c}\circ \\ \ell=1 \quad i=0 \\ n=1\end{array}}$$

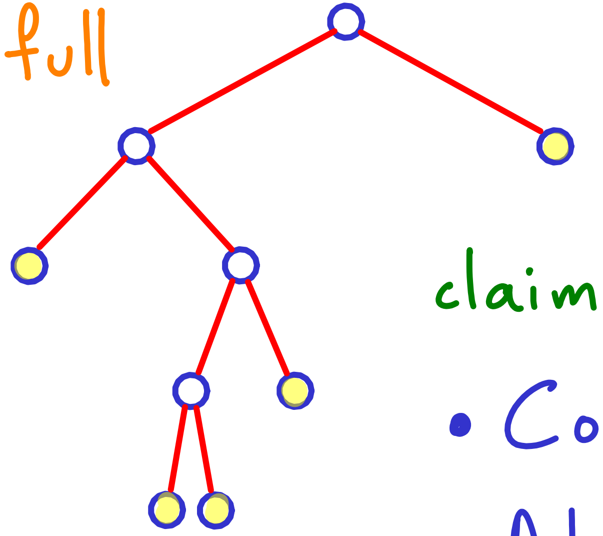
ℓ = #internal nodes $+1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.
- New tree has $i-1$ internal nodes & is full.
  - ↳ has i leaves by hypothesis.
- Put x & y back. $+2$ leaves
- $p \rightarrow$ internal : -1 leaf

full



$\ell = \#\text{internal nodes} + 1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $< i$ internal nodes.

base case trivial

$\boxed{\ell = 1 \quad i = 0 \quad n = 1}$

- Consider any full tree with i internal nodes.
- At lowest level, $\exists$ leaf x.
- x has a sibling y. (parent p of x has 2 children)
- Remove x & y : p becomes a leaf.
- New tree has $i-1$ internal nodes & is full.

  $\hookrightarrow$ has i leaves by hypothesis.
- Put x & y back. $+2$ leaves
- $p \rightarrow$ internal : -1 leaf

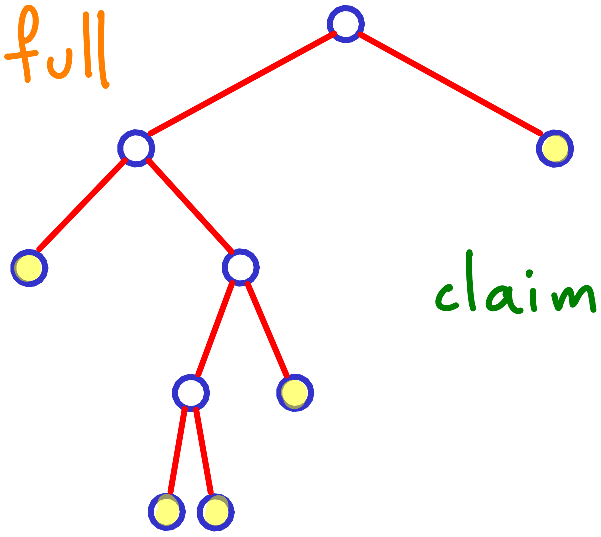
$\left.\begin{matrix} +2 \text{ leaves} \\ -1 \text{ leaf} \end{matrix}\right\} i+1$ leaves

$\square$

Alternate proof

(for practice)

full



ℓ = #internal nodes + 1 = i + 1

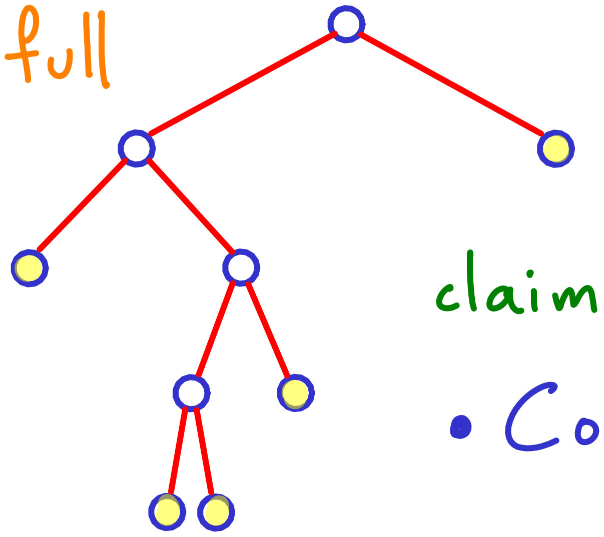
Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

base case
trivial

$\ell = 1$ $\quad i = 0$

$n = 1$



ℓ = # internal nodes $+1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.

base case
trivial

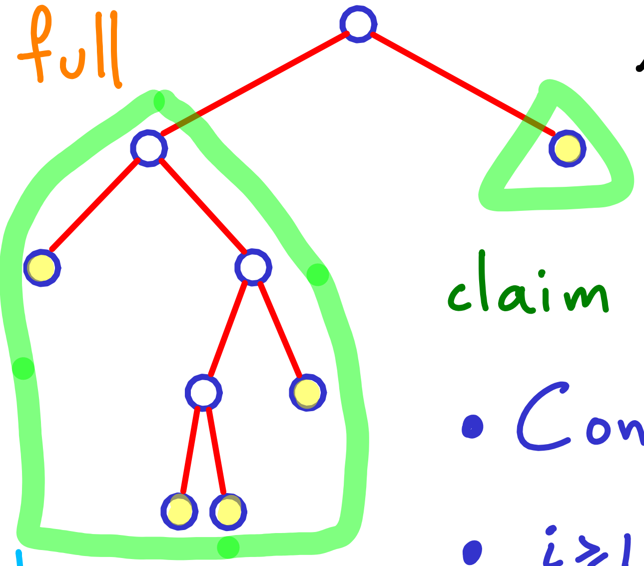
$\ell = 1 \quad i = 0$

$n = 1$

$\ell = \#\text{internal nodes} + 1 = i+1$

Induction on i

full



base case

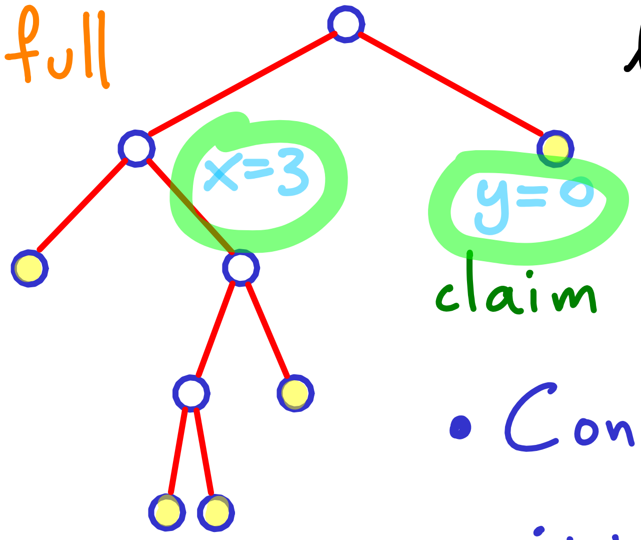
trivial

$\ell=1$ $i=0$

$n=1$

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $<i$ internal nodes.

- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.



ℓ = #internal nodes $+1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis: claim holds for full trees with $<i$ internal nodes.

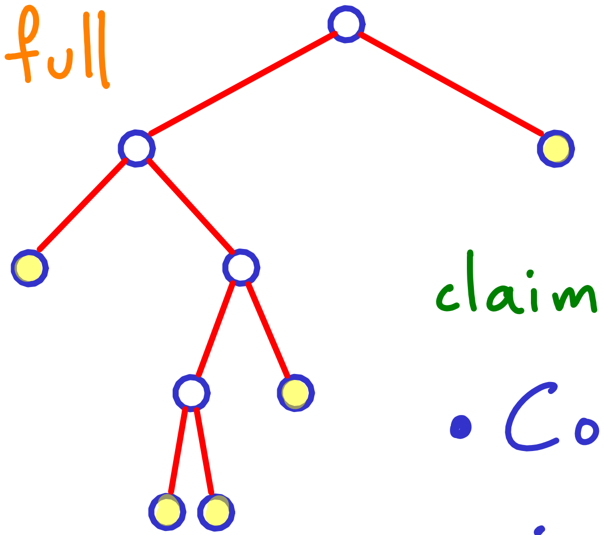
- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.

base case trivial

$\ell=1$ $i=0$

$n=1$

full



ℓ = #internal nodes +1 = i+1

Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

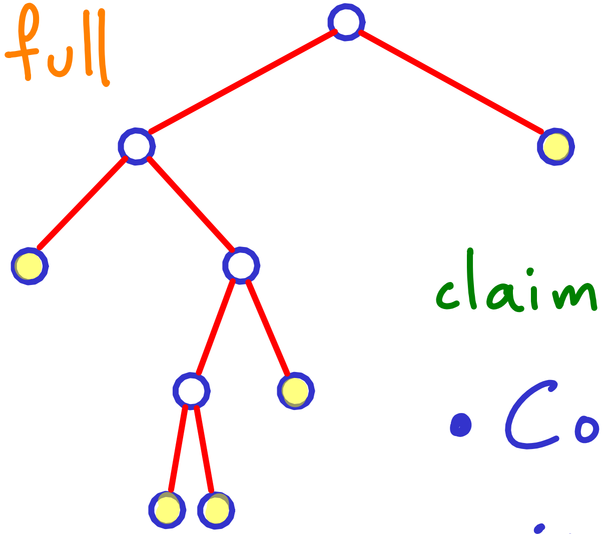
- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.
- $x+y = i-1$ (excluding root)

base case
trivial

$\ell=1 \quad i=0$

$n=1$

full



$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

Induction on i

base case
trivial

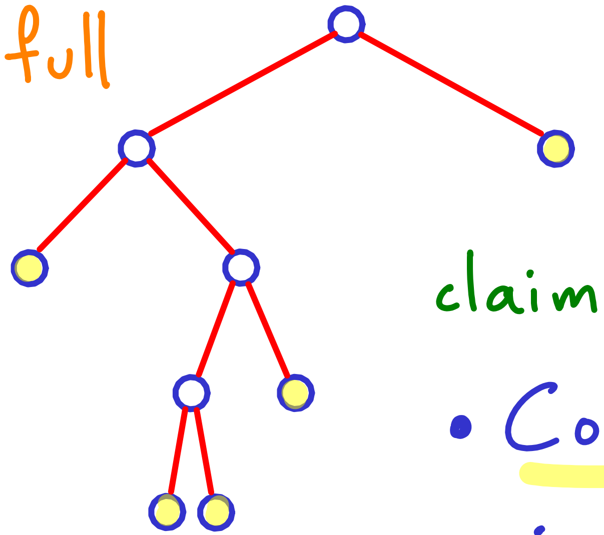
$\ell = 1 \quad i = 0$

$n = 1$

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.
- $x + y = i - 1$ (excluding root) $\rightarrow$ $x \leq i-1$ & $y \leq i-1$

full



base case trivial

$\bigcirc$

$\ell = 1 \quad i = 0$

$n = 1$

$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

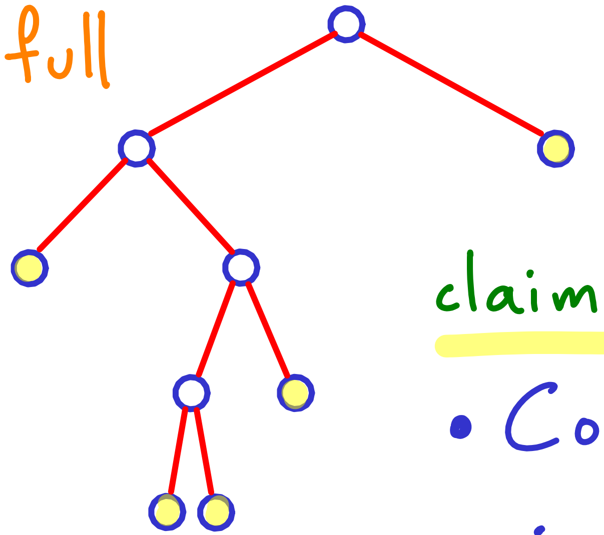
Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.
- $x + y = i - 1$ (excluding root) → $x \leq i-1$ & $y \leq i-1$
- Subtrees are full

→ by contradiction

full



base case
trivial

$\ell = 1 \quad i = 0$
$n = 1$

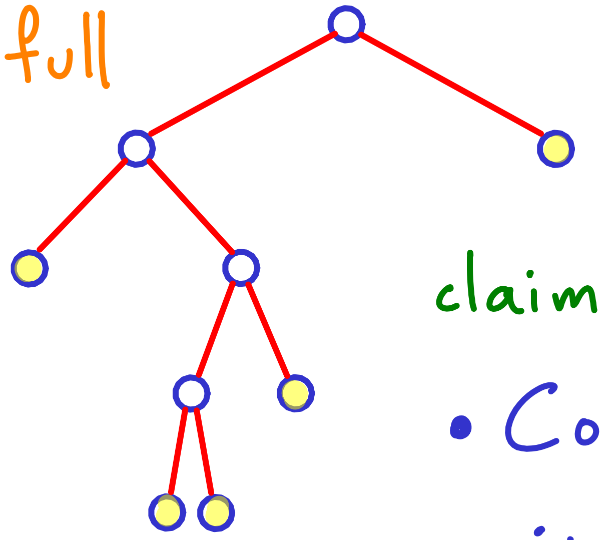
$\ell = \underline{\#\text{internal nodes}} + 1 = i + 1$

Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.
- $x + y = i - 1$ (excluding root) $\rightarrow$ $x \leq i-1$ & $y \leq i-1$
- Subtrees are full, so by hypothesis,
  left subtree has $x+1$ leaves & right subtree has $y+1$.

full



base case trivial

$\circ$

$\ell=1$ $i=0$

$n=1$

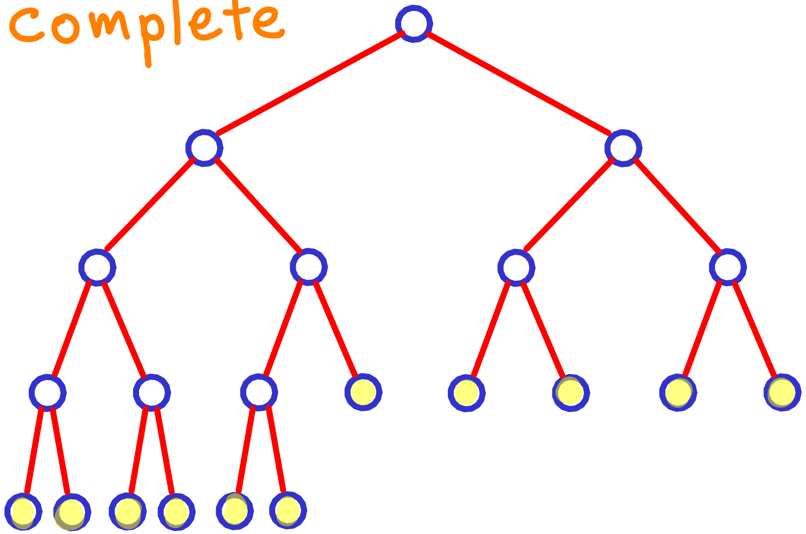
$\ell = \underline{\#\text{internal nodes}} + 1 = i+1$

Induction on i

To prove for $i \geq 1$, hypothesis:
claim holds for full trees with $< i$ internal nodes.

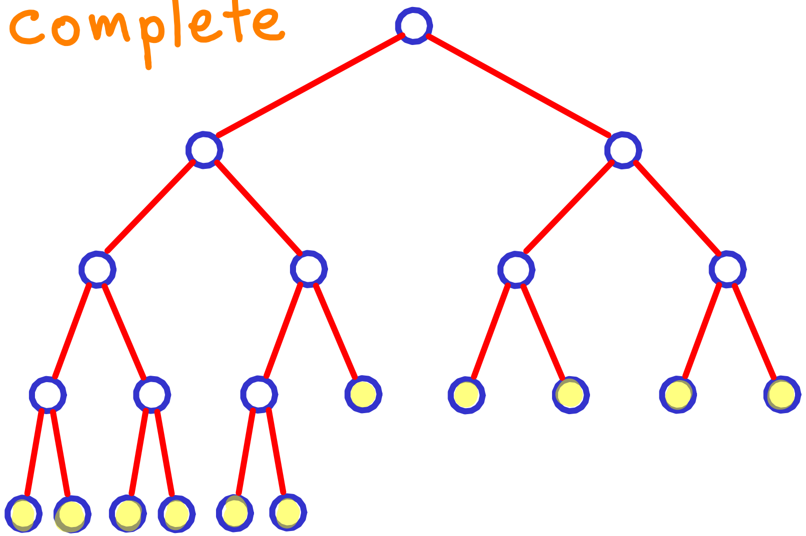
- Consider any full tree with i internal nodes.
- $i \geq 1$ so the root has 2 non-empty subtrees.
- Left subtree has x internal nodes, right subtree has y.
- $x+y = i-1$ (excluding root) $\rightarrow$ $x \leq i-1$ & $y \leq i-1$
- Subtrees are full, so by hypothesis,
  left subtree has $x+1$ leaves & right subtree has $y+1$.
- Combined, they have $x+y+2 = (i-1)+2 = i+1$ leaves $\square$

complete



n is odd, tree is full.

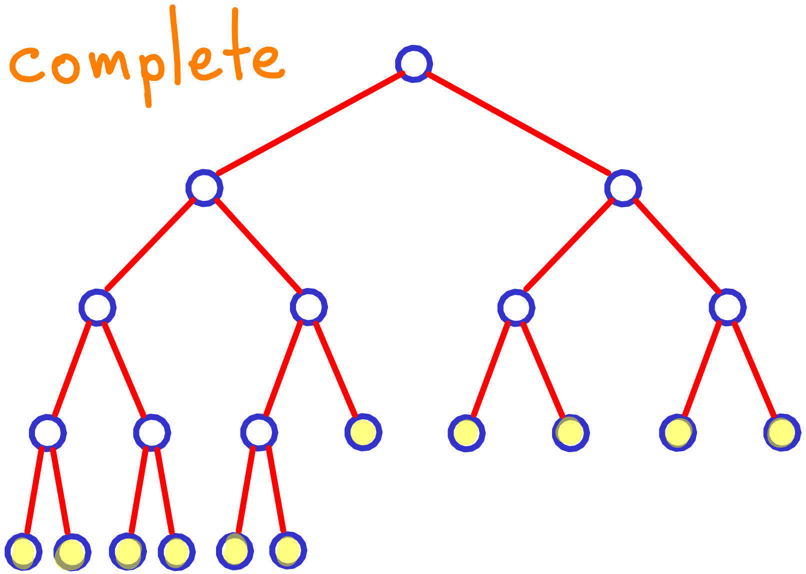
complete



n is odd, tree is full.

ℓ = # internal nodes +1

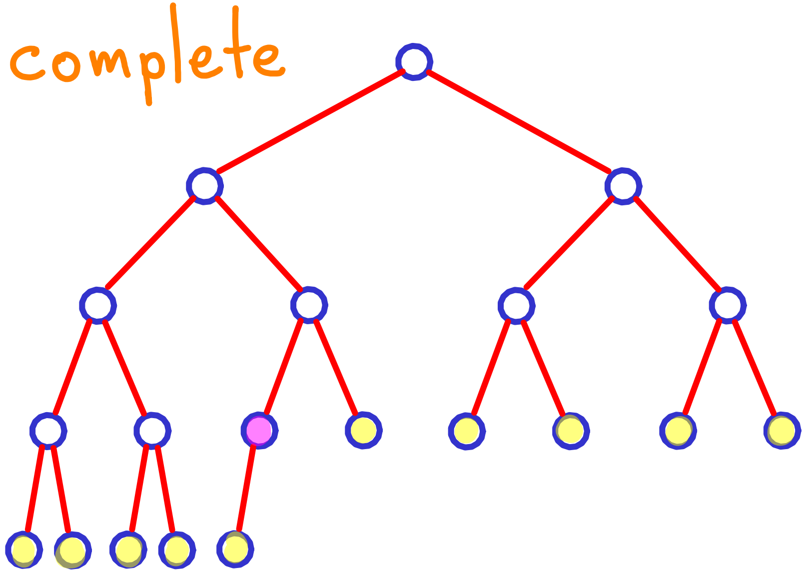
$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$



n is odd, tree is full.

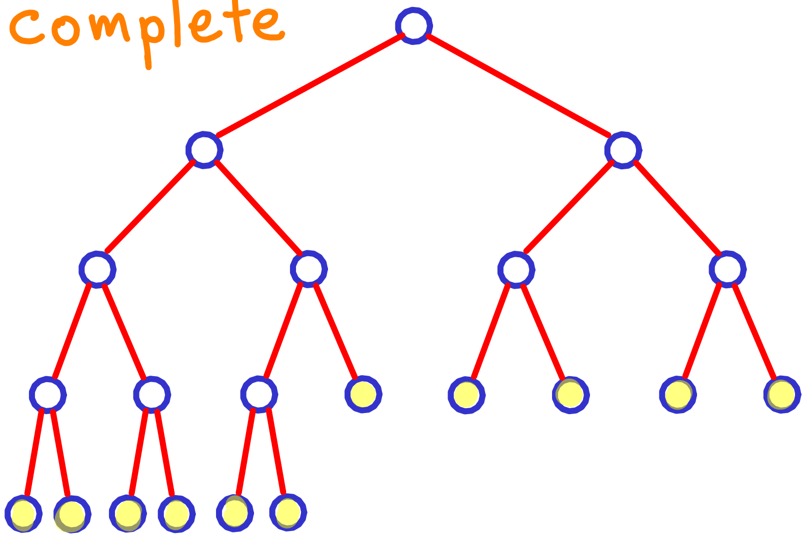
ℓ = # internal nodes + 1

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$



n is even, tree is not full.

complete

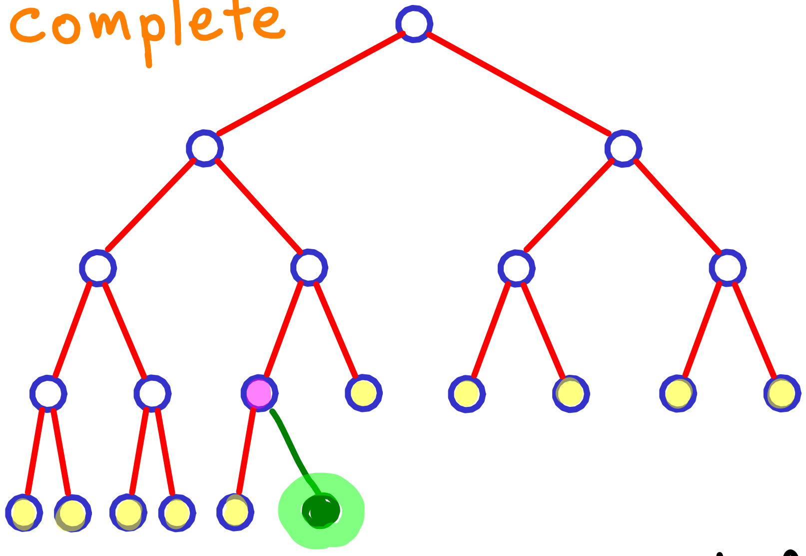


n is odd, tree is full.

$$\ell = \#\text{internal nodes} + 1$$

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

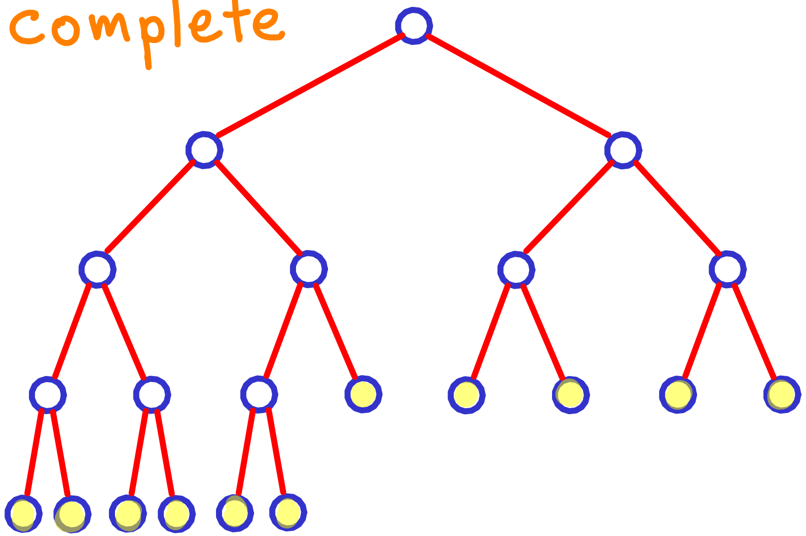
complete



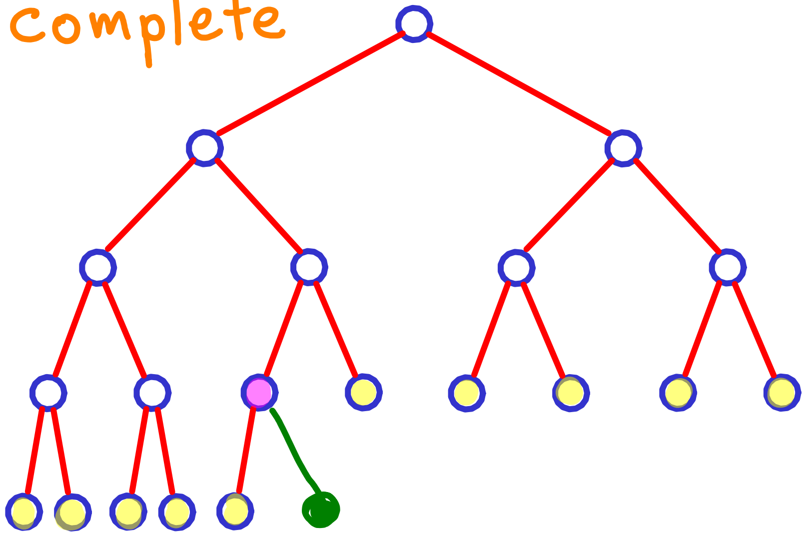
n is even, tree is not full.

- can add one leaf...

complete



complete



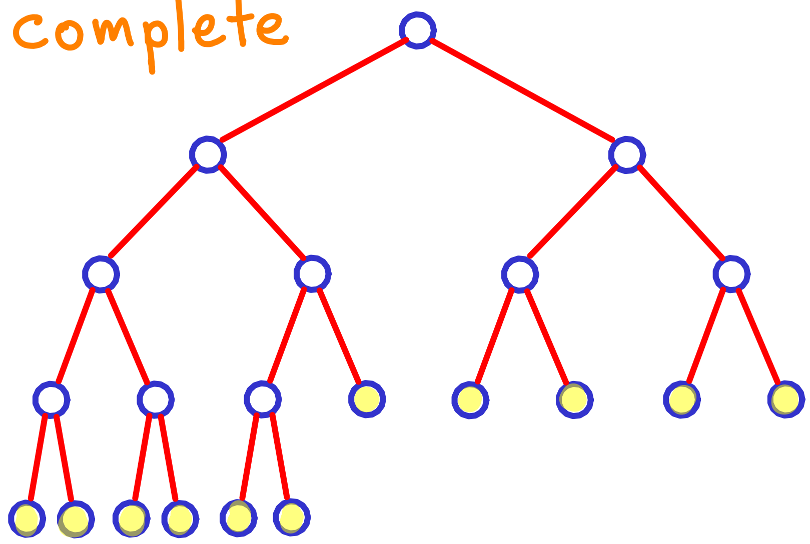
n is odd, tree is full.

$$\ell = \#\text{internal nodes} + 1$$

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

n is even, tree is not full.

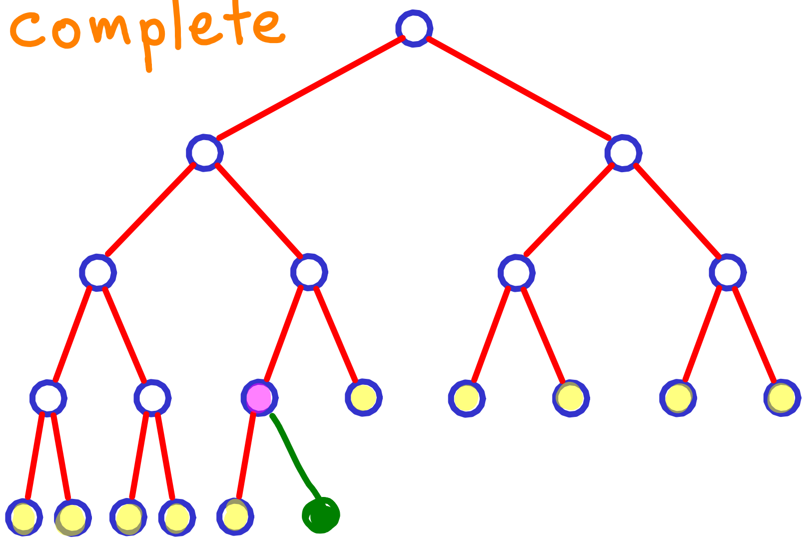
- can add one leaf...
- new tree has n+1 nodes & is full



n is odd, tree is full.

ℓ = # internal nodes + 1

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

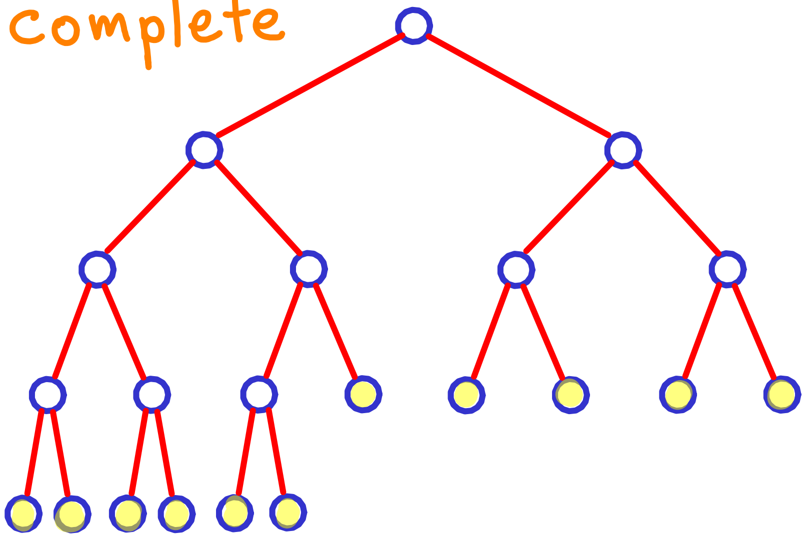


n is even, tree is not full.

- can add one leaf...
- new tree has n+1 nodes & is full

$$\ell + 1 = \frac{(n+1)+1}{2}$$

complete

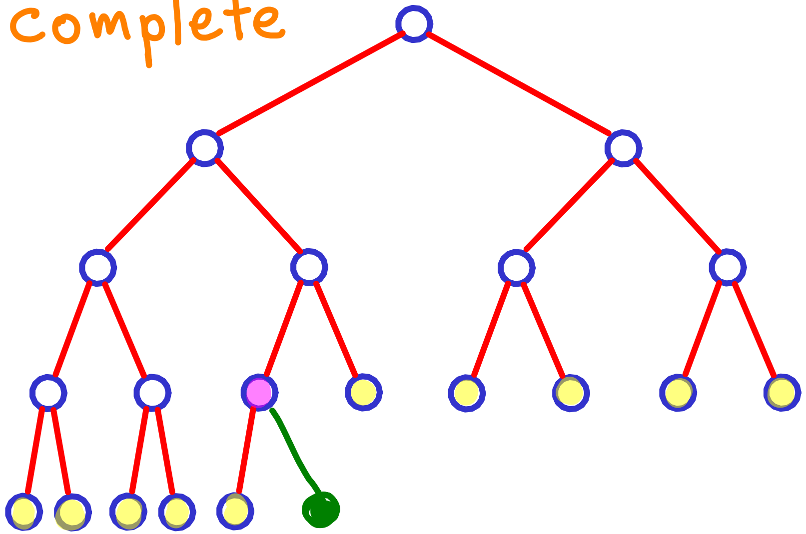


n is odd, tree is full.

ℓ = # internal nodes + 1

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

complete

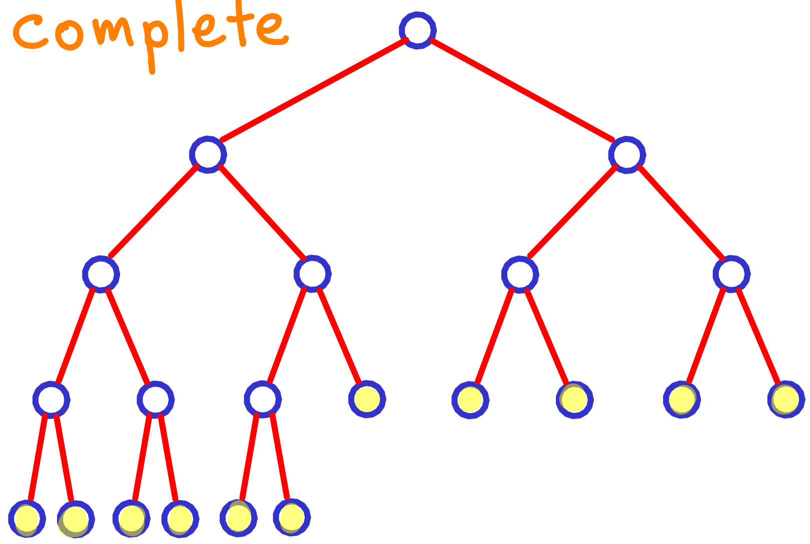


n is even, tree is not full.

- can add one leaf...
- new tree has n+1 nodes & is full

$$\ell + 1 = \frac{(n+1)+1}{2} \rightarrow \ell = \frac{n}{2}$$

complete

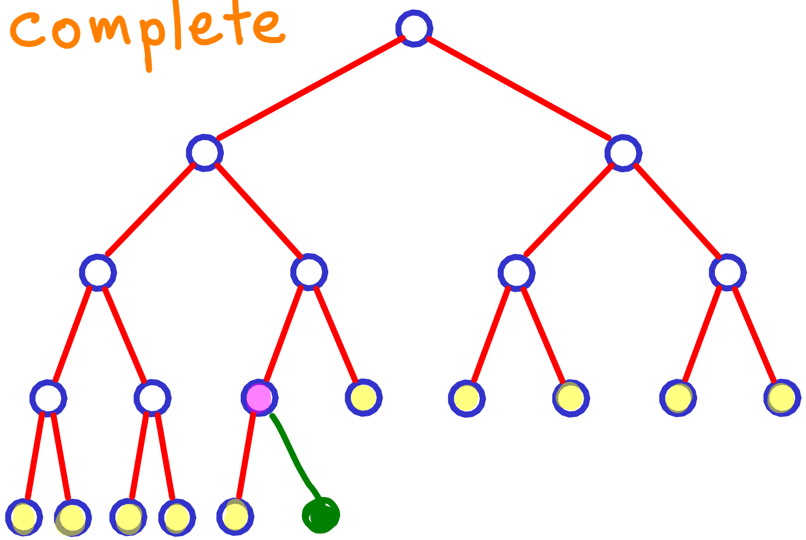


n is odd, tree is full.

ℓ = #internal nodes +1

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

complete

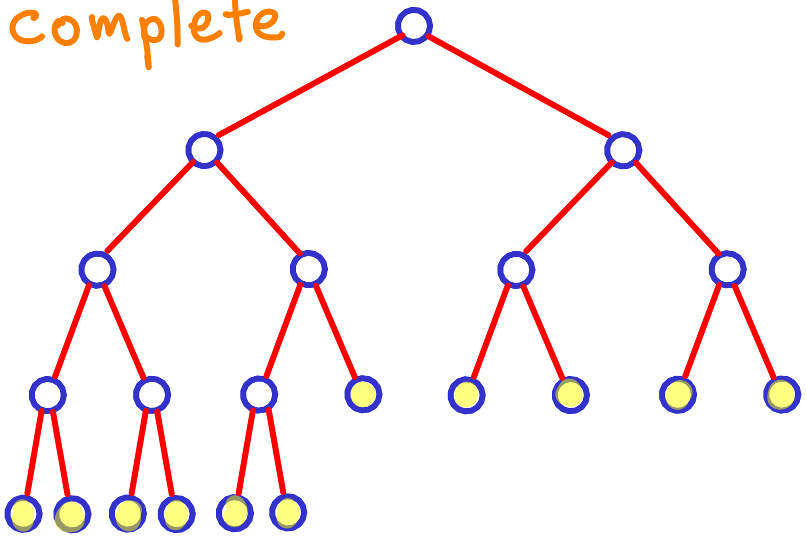


n is even, tree is not full.

- can add one leaf...
- new tree has n+1 nodes & is full

$$\ell + 1 = \frac{(n+1)+1}{2} \rightarrow \ell = \frac{n}{2} = \left\lceil \frac{n}{2} \right\rceil$$

complete

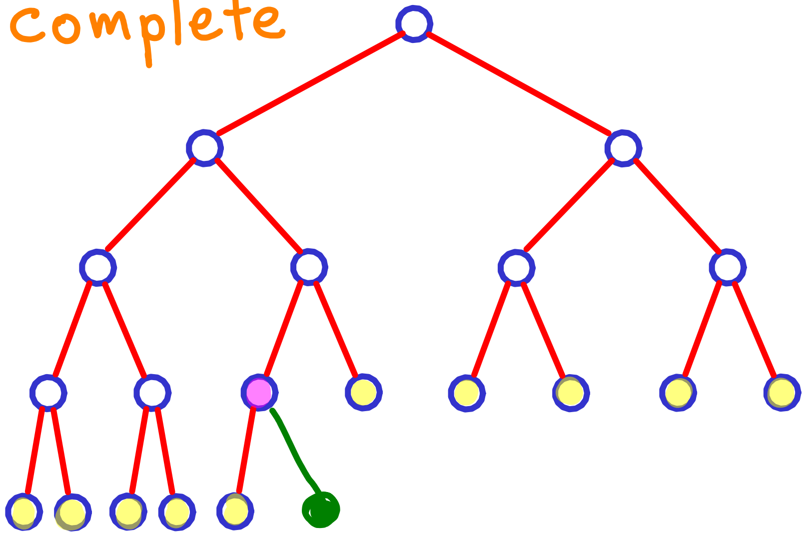


n is odd, tree is full.

ℓ = #internal nodes +1

$$\ell = \left\lceil \frac{n}{2} \right\rceil = \frac{n+1}{2}$$

complete



n is even, tree is not full.

- can add one leaf...
- new tree has n+1 nodes & is full

$$\ell + 1 = \frac{(n+1)+1}{2} \rightarrow \ell = \frac{n}{2} = \left\lceil \frac{n}{2} \right\rceil$$

ℓ = #internal nodes

For perfect binary trees, we saw:

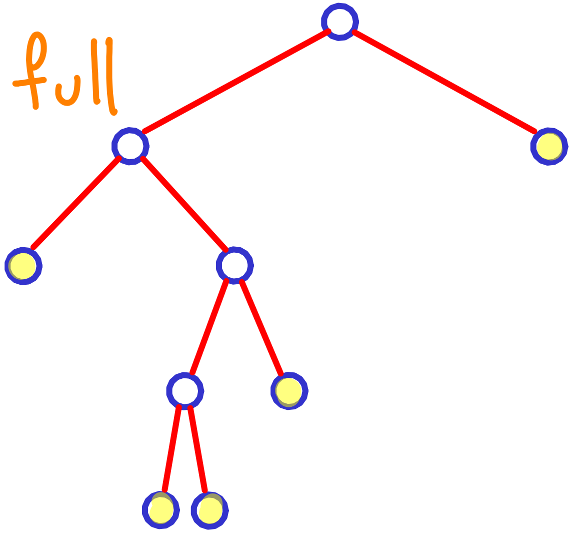
ℓ vs h

$$\ell = 2^h$$

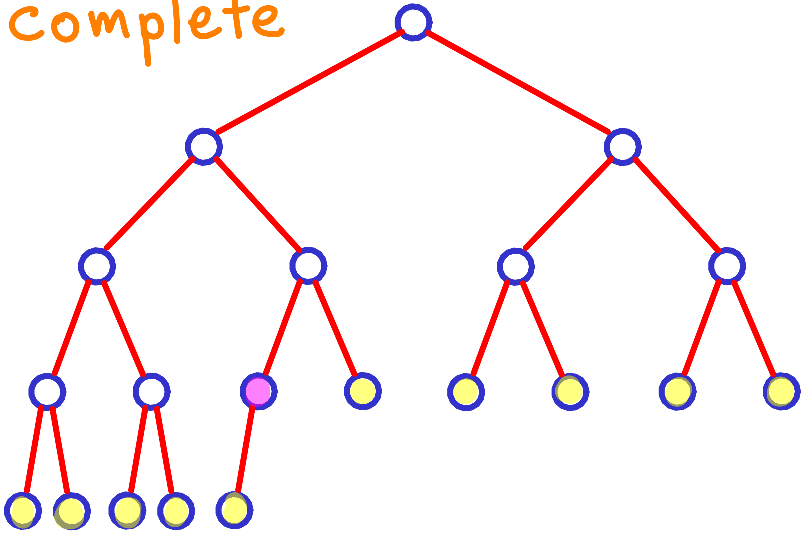
$$h = \log_2 \ell$$

What about:

full

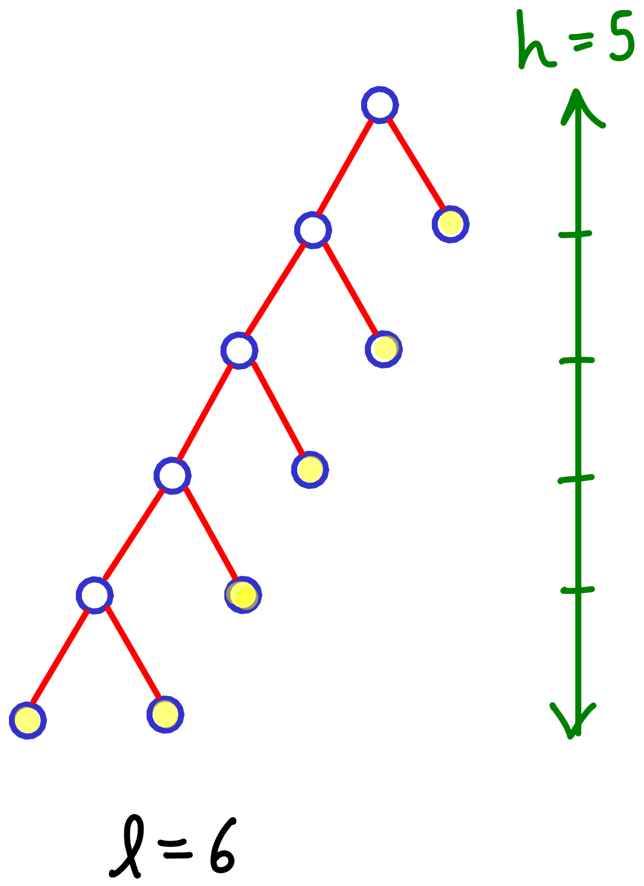


complete



full

can get $\ell = h+1$



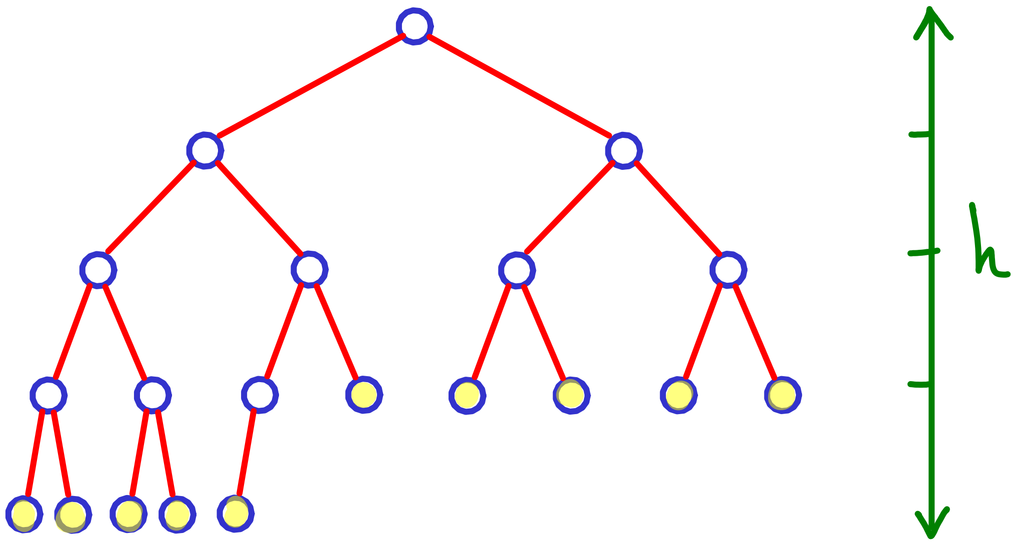
$\ell \neq 2^h$

$h \neq \log_2 \ell$

not even $h = \Theta(\log \ell)$

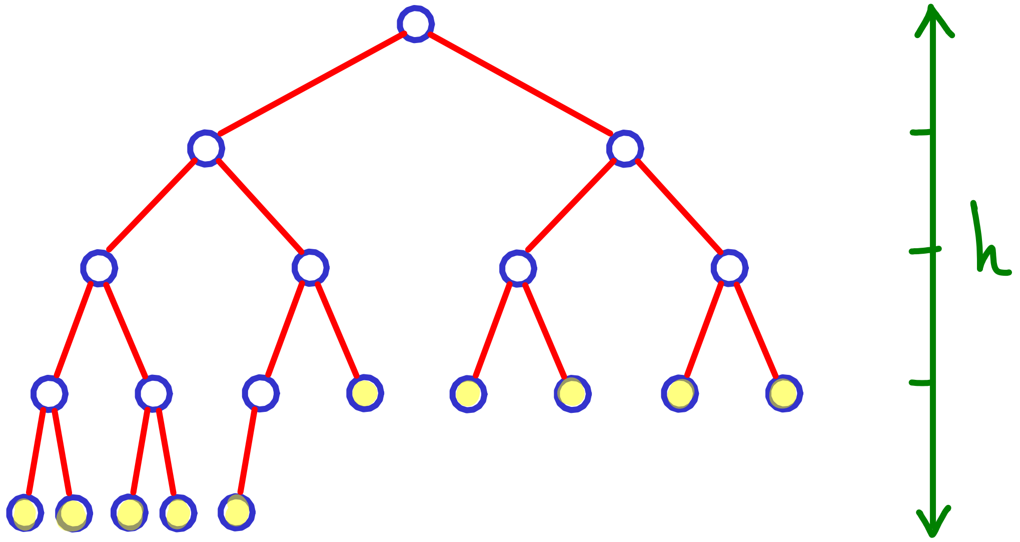
complete

n nodes → what is h ?



complete
n nodes → what is h?

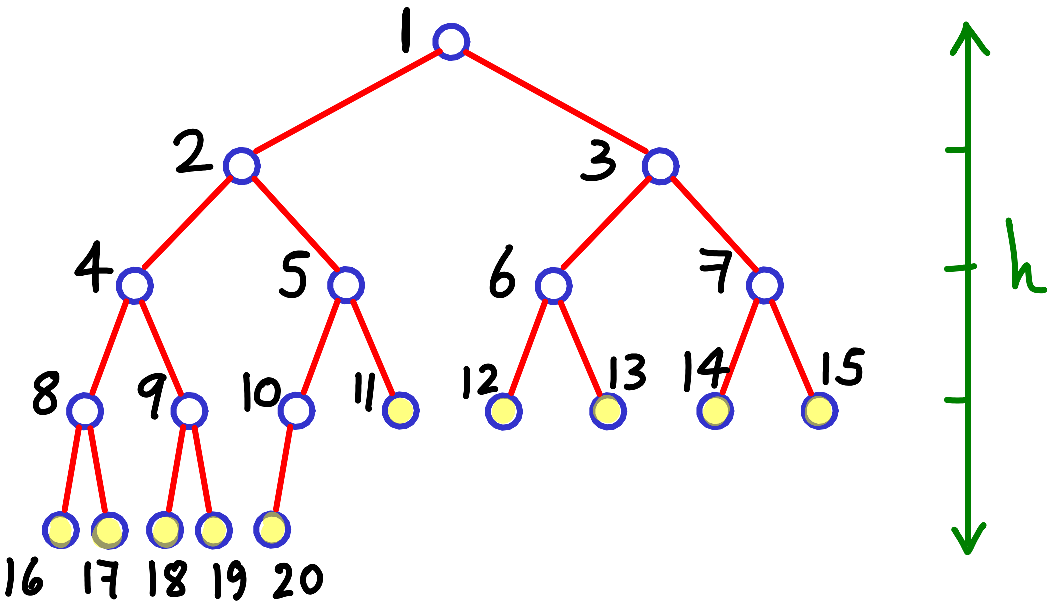
Intuition: h doesn't change much compared to perfect tree



complete

n nodes → what is h?

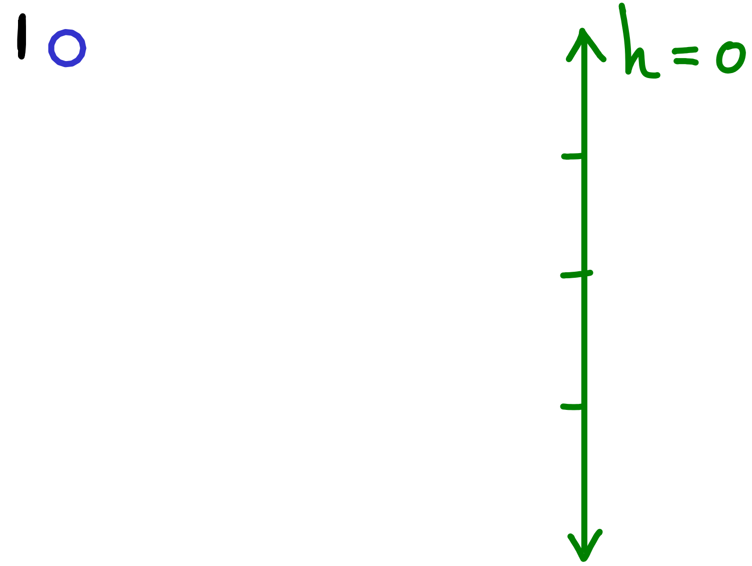
Intuition: h doesn't change much compared to perfect tree



complete

n nodes → what is h?

Intuition: h doesn't change much compared to perfect tree

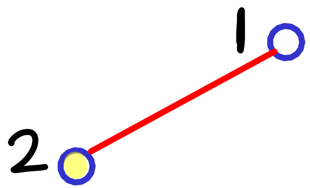


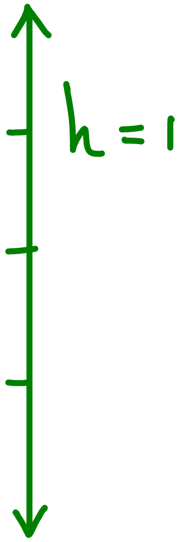
$h = \log_2 n$

complete

n nodes → what is h?

Intuition: h doesn't change much compared to perfect tree

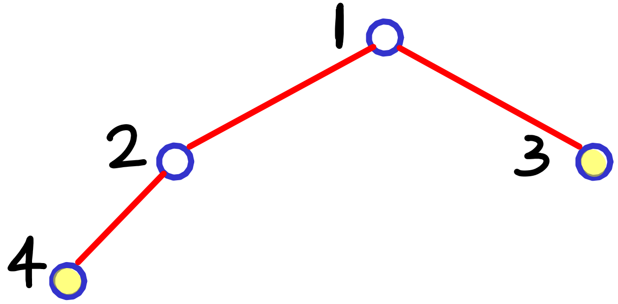




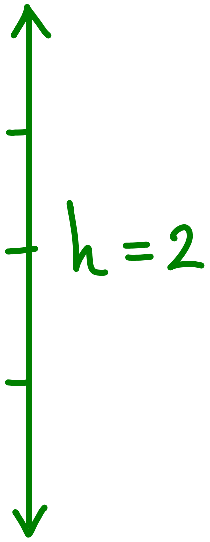
$h = \log_2 n$

complete

n nodes → what is h ?



Intuition: h doesn't change much compared to perfect tree

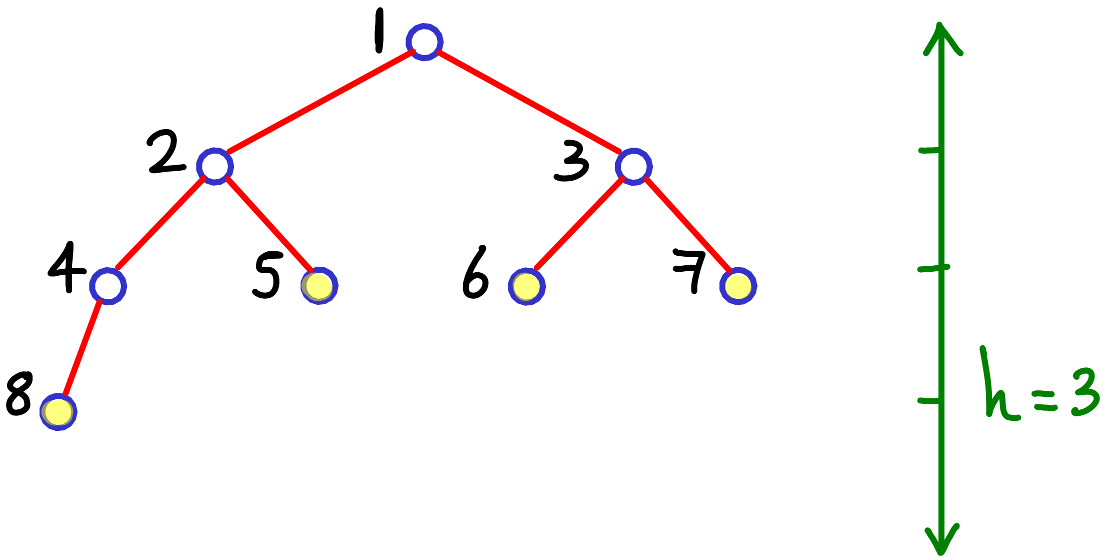


$h = \log_2 n$

complete

n nodes → what is h?

Intuition: h doesn't change much compared to perfect tree

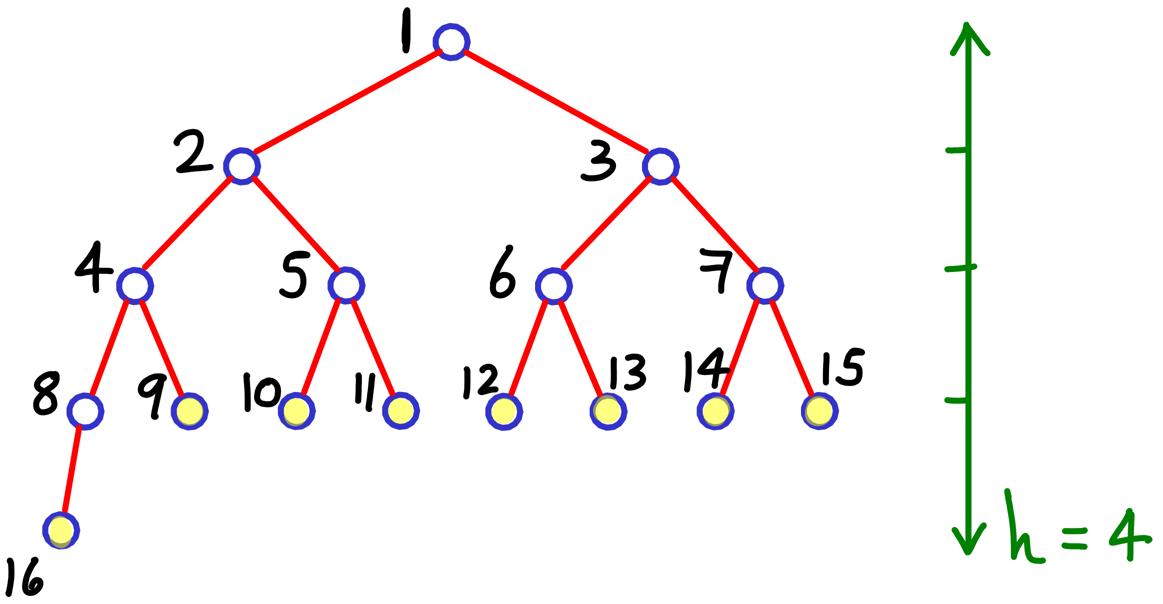


$h = \log_2 n$

complete

n nodes → what is h ?

Intuition: h doesn't change much compared to perfect tree

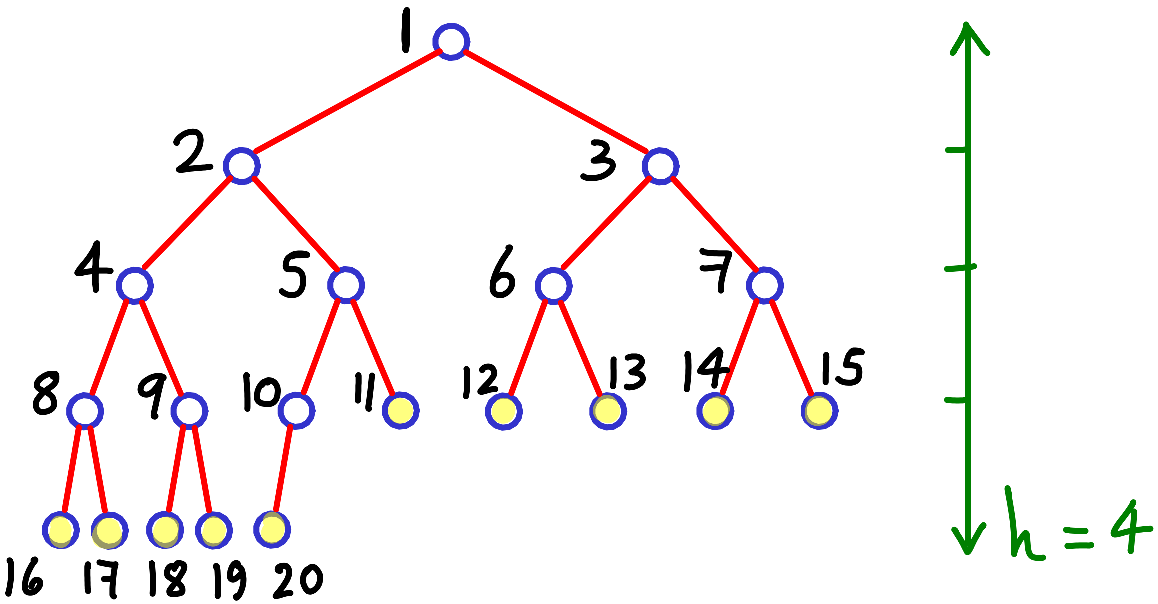


$h = \log_2 n$

complete

n nodes → what is h ?

Intuition: h doesn't change much compared to perfect tree



for $2^h < n < 2^{h+1}$

$h < \log_2 n < h+1$

$h = \lfloor \log_2 n \rfloor$

balanced binary trees

balanced binary trees

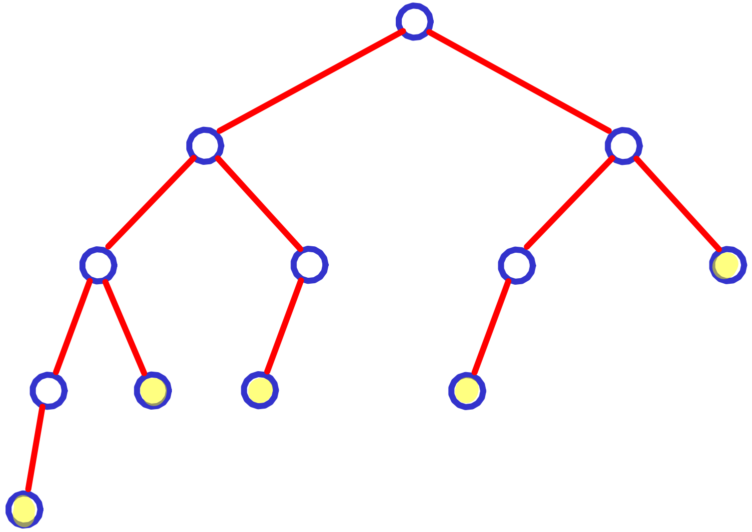
no unique definition

common: depth of left & right subtrees differ by ≤ 1

balanced binary trees

no unique definition

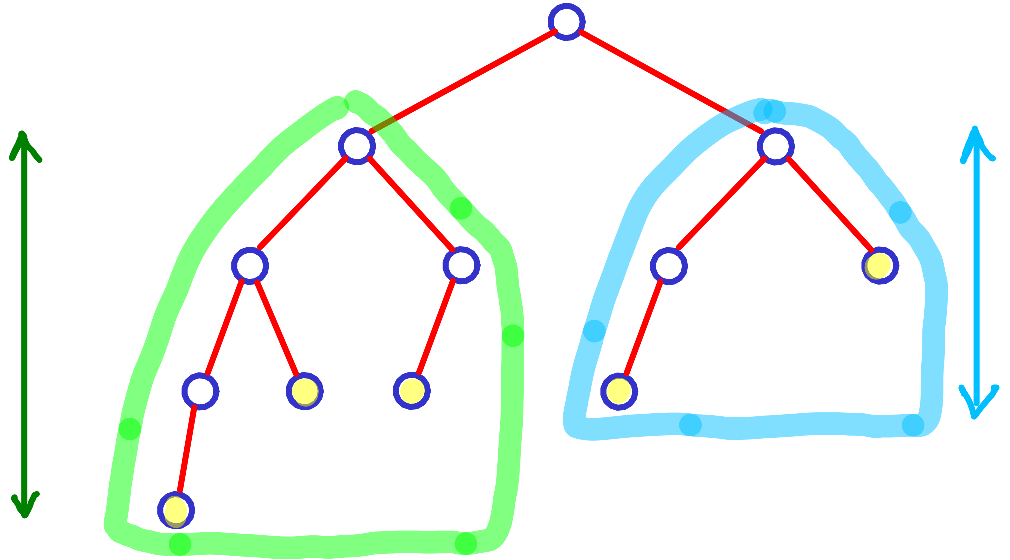
common: depth of left & right subtrees differ by ≤ 1



balanced binary trees

no unique definition

common: depth of left & right subtrees differ by ≤ 1

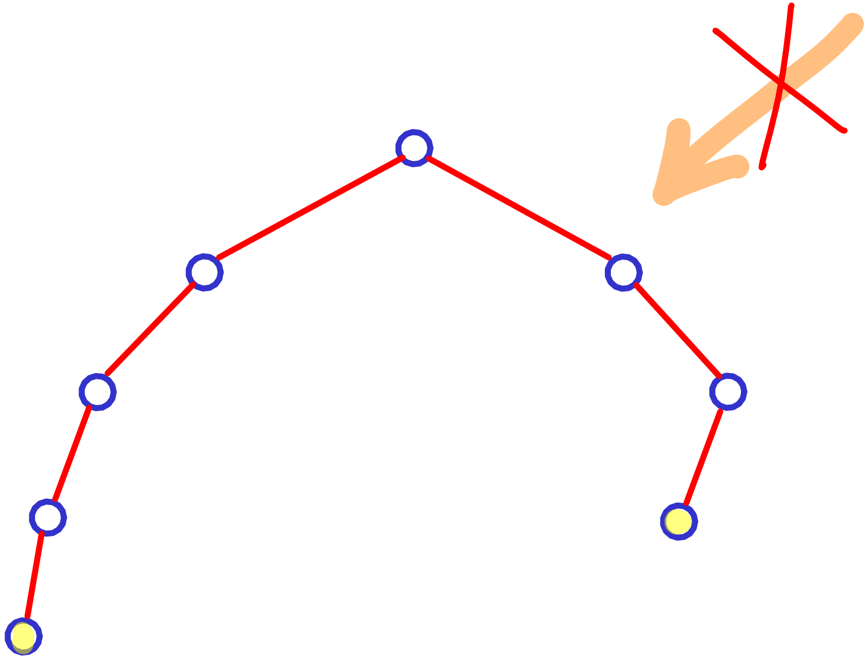


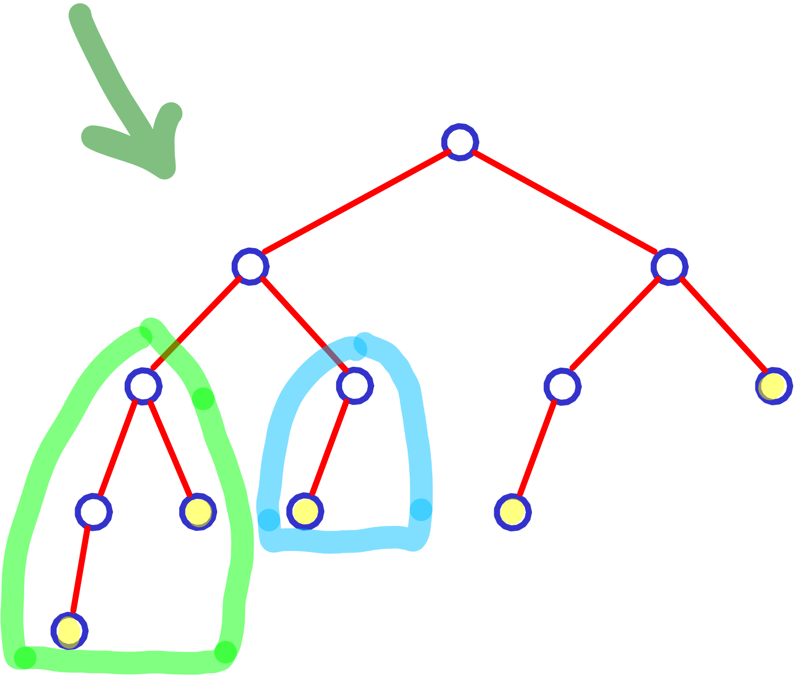
balanced binary trees

no unique definition

common: depth of left & right subtrees differ by ≤ 1

(recursively)



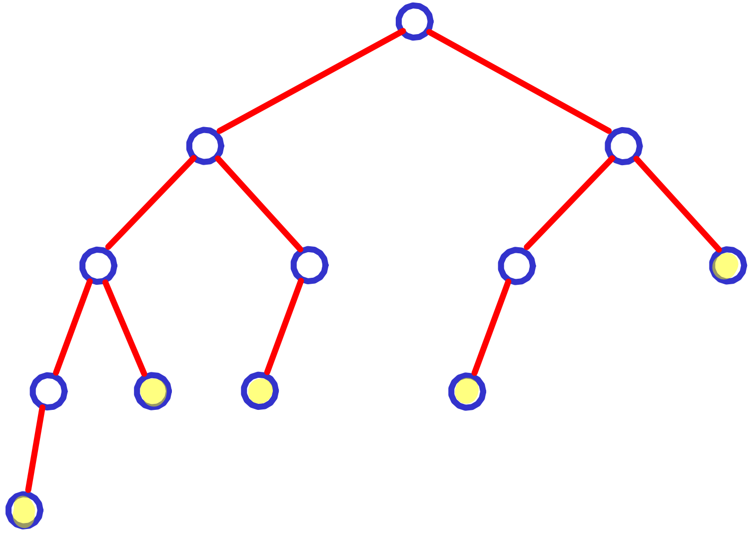


balanced binary trees

no unique definition

common: depth of left & right subtrees differ by ≤ 1 (recursively)

also common: depth is $O(\log n)$



balanced binary trees

no unique definition

common: depth of left & right subtrees differ by ≤ 1
(recursively)

implies

doesn't imply

also common: depth is $O(\log n)$