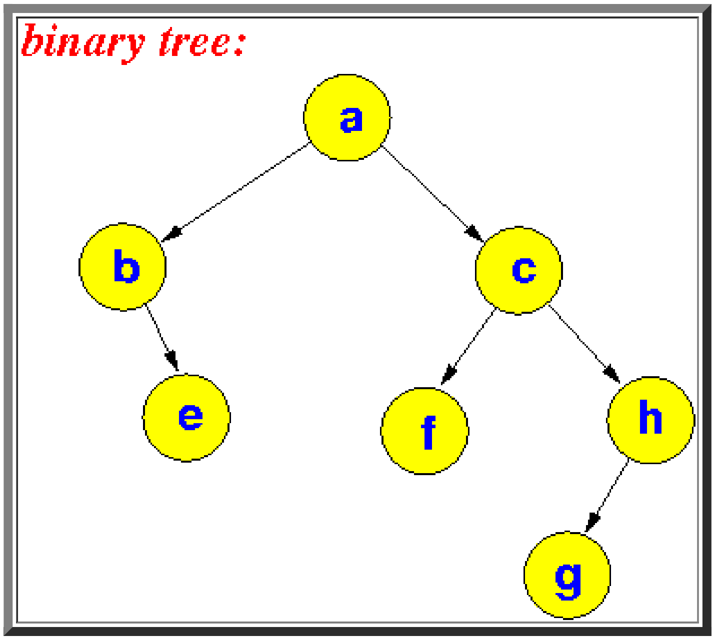
Binary trees

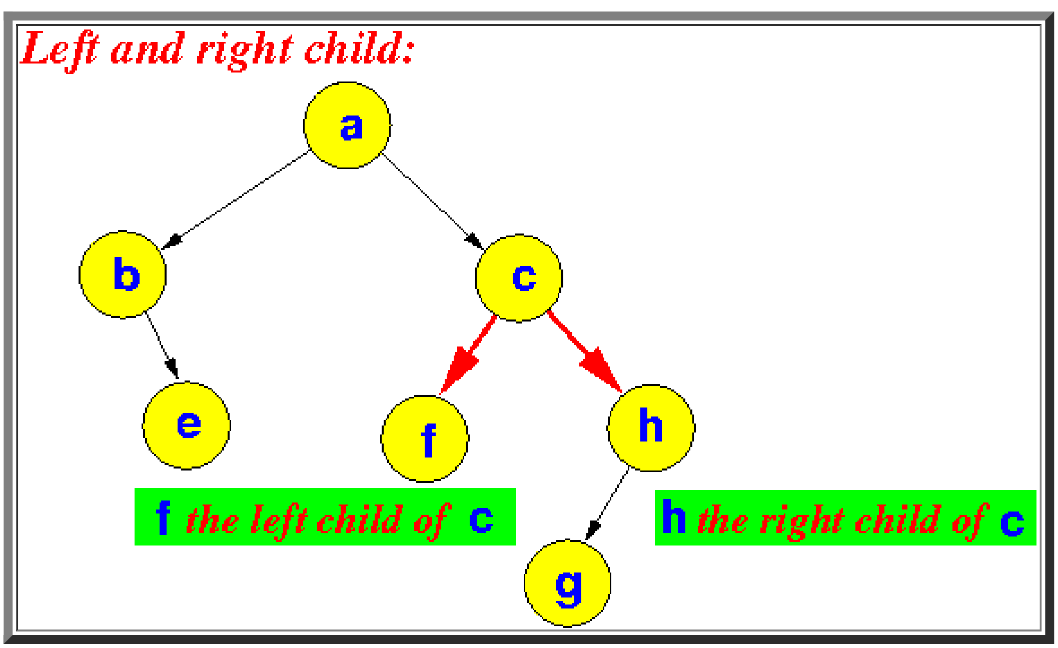
- **Binary tree**
  - ***Binary*** **tree**:

    - **Binary tree** = a **tree** where ***each*** **node** has ***at most*** **2 children nodes**

    Example:



- **Left and right child**
  - Because **each node** has ***at most*** **2** children nodes, we can **label** the children **distinctly** as **left** and **right**:



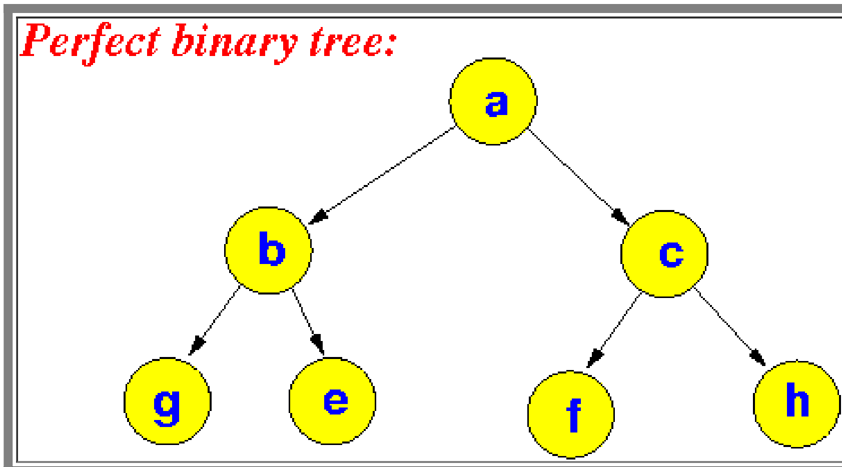
Note:

- Some nodes (e.g. ***h***) have ***only*** a ***left* child node**
- Some nodes (e.g. ***b***) have ***only*** a ***right* child node**

- **Perfect binary tree**
  - **Perfect** binary tree
    - **Perfect binary tree** = a **binary tree** where **each level** contains the ***maximum* number of nodes**

      I.e., **every level** is **completely *full*** of nodes

    Example:



- **Some properties of the *perfect* binary tree**
  - **Property 1:**
    - The **number of nodes** at **depth *d*** in a ***perfect* binary tree** = 2^d

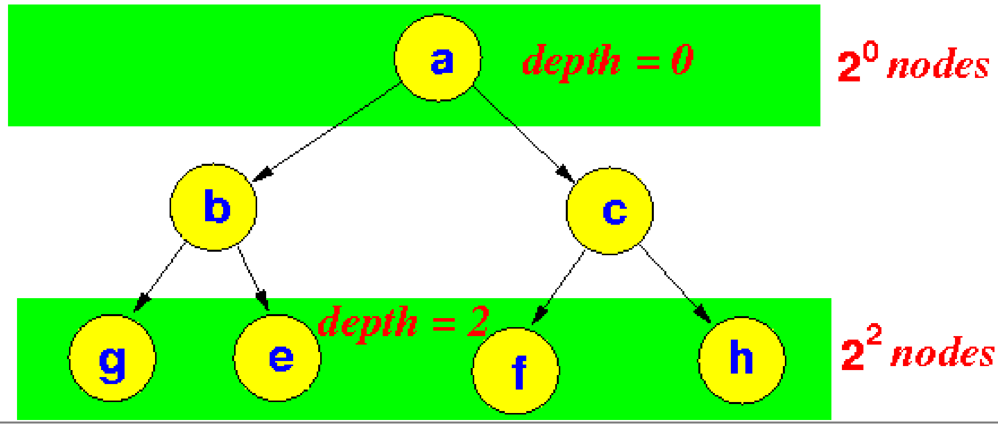
    Proof:

    - There is ***only*** **1 node** (= the **root node**) at **depth *0***:

      $2^0 = 1$

    - In a **perfect binary tree**, **every node** has **2 children nodes**

Number of nodes at depth d:



So:

Depth d	# nodes at depth d		# of child nodes
0	1	$= 2^0$	2 (each node has 2 children)
1	2	$= 2^1$	4 (each node has 2 children)
2	4	$= 2^0$	8 (each node has 2 children)
...			

I.e.:

- The **number** of nodes **doubles** every time the **depth** increases by **1** !

Therefore:

`# nodes at depth d =` 2^d

- **Property 2:**

  - A ***perfect* binary tree** of **height *h*** has:

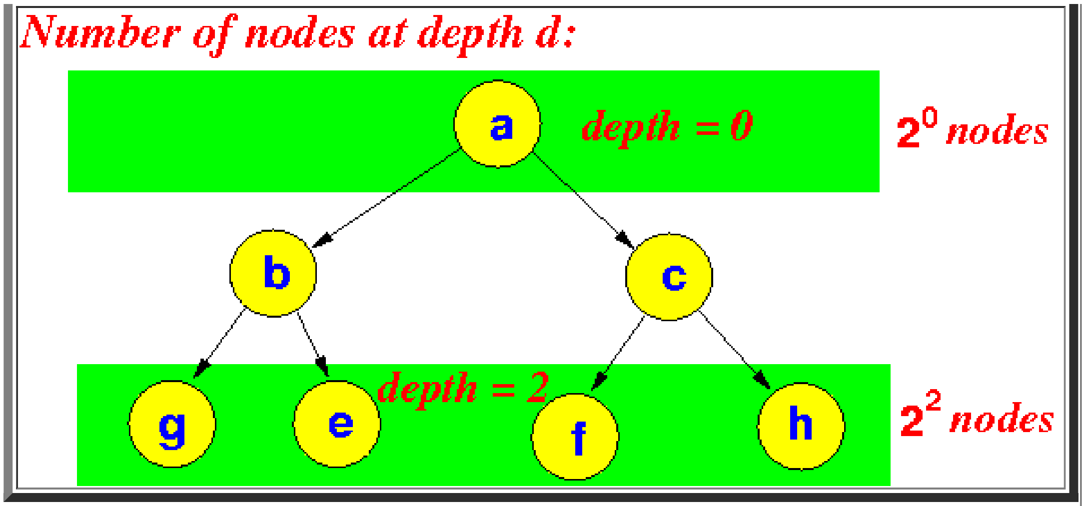
    - $2^{h+1} - 1$ nodes

Proof:

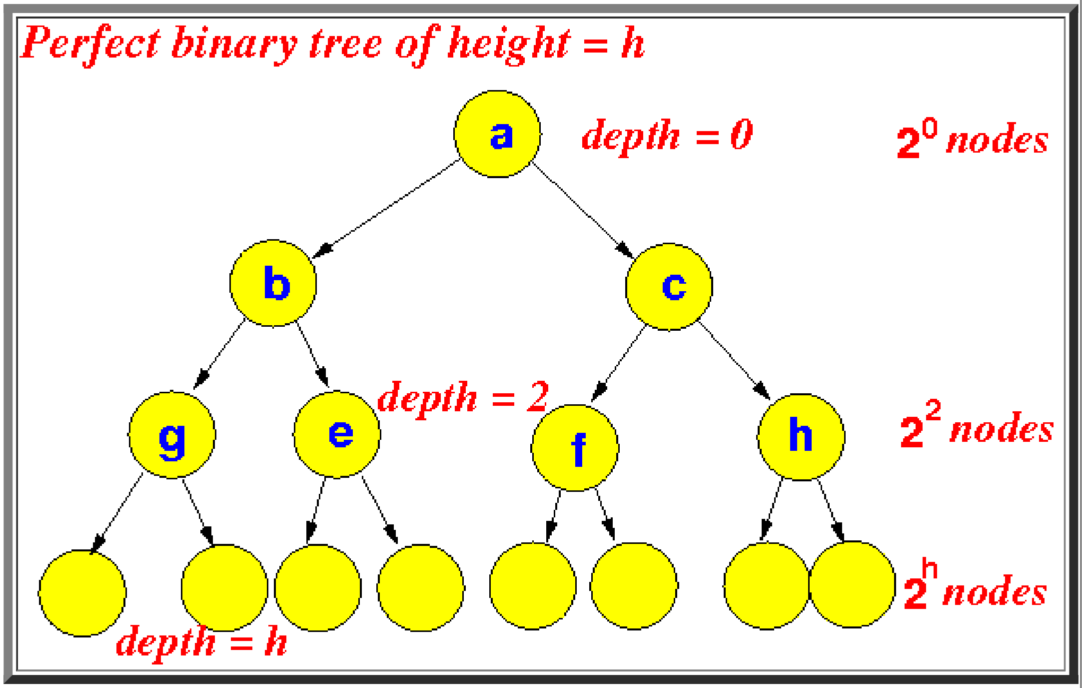
- **Previously,** we have shown that:

  - # nodes at ***depth* d** = 2^d

  See:



- So the **total number** of **nodes** in a **perfect binary tree** of **height *h***:



nodes = $2^0 + 2^1 + ... 2^h = 2^{h+1} - 1$

- **Proof:**

```
        S = 1 + 2 + 2^2 + 2^3 + ... + 2^h
      2xS =     2 + 2^2 + 2^3 + ... + 2^h + 2^(h+1)    - (subtract)
  ------------------------------------------------------------
   2xS - S = 2^(h+1) - 1
<==>     S = 2^(h+1) - 1
```

- **Property 3:**

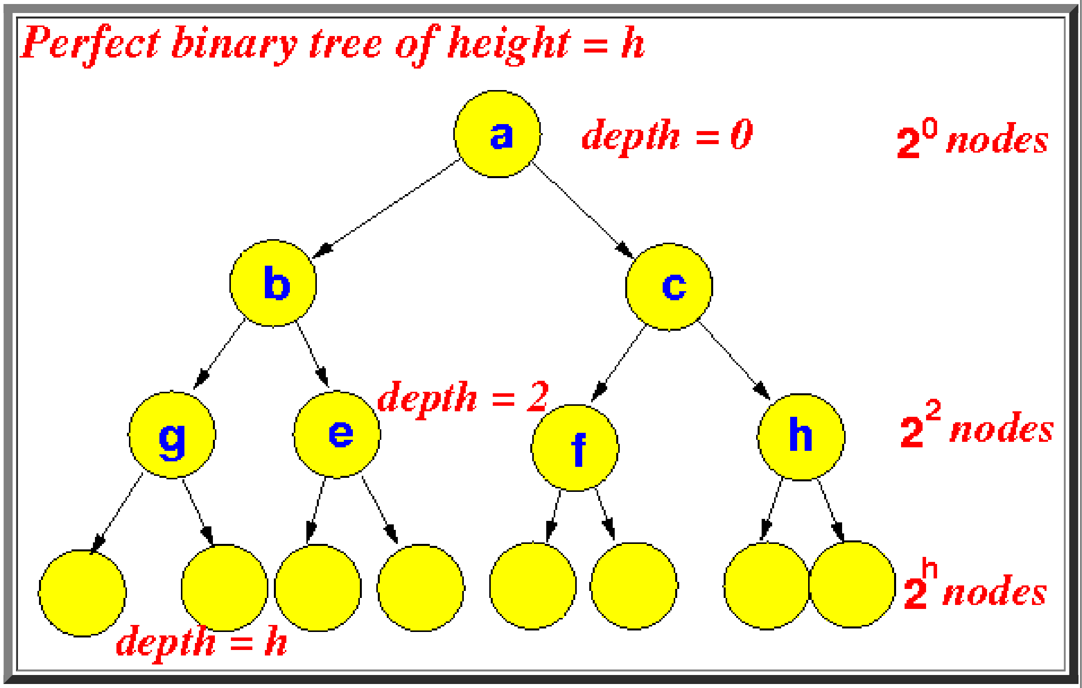
- **Number** of ***leaf* nodes** in a **perfect binary tree** of **height *h*** = 2^h

Proof:

- # nodes at **depth *d*** in a **perfect binary tree** = 2^d

---

- **All** the ***leaf* nodes** in a **perfect binary tree** of **height *h*** has a depth equal to ***h***:



- # nodes at **depth *h*** in a **perfect binary tree** = 2^h

Therefore:

- **Number** of ***leaf* nodes** in a **perfect binary tree** of **height *h*** = 2^h

---

- **Property 4:**

- **Number** of ***internal* nodes** in a **perfect binary tree** of **height *h*** = $2^h - 1$

Proof:

- # nodes in a **perfect binary tree** of **height *h*** = $2^{h+1} - 1$ (see **Property 2**)

- # ***leaf*** nodes in a **perfect binary tree** of **height *h*** = 2^h (see **Property 3**)

- The **other nodes** are ***internal* nodes** (i.e., with at least 1 child node).

So:

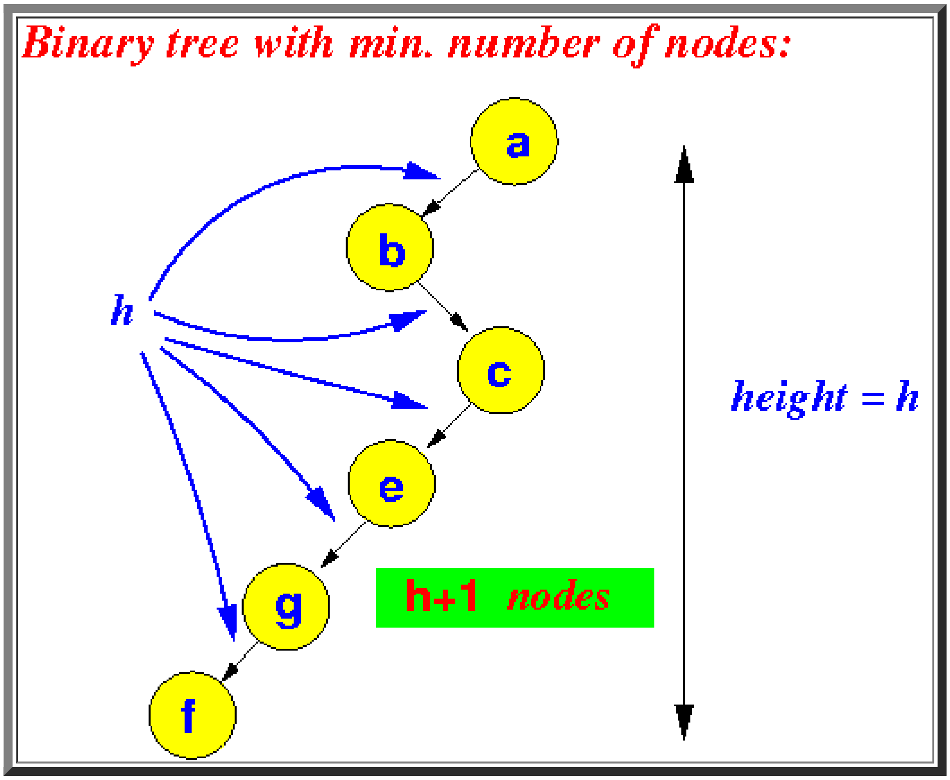
- # **internal** nodes in a **perfect binary tree** of **height** ***h*** = $(2^{h+1} - 1) - 2^h = 2^h - 1$

- **Minimum and maximum number of nodes in a binary tree of height *h***
  - **Fact:**

    - The ***minimum*** **number of nodes** in a **binary tree** of ***height*** **h** = ***h*** **+ 1**

    Proof:

    - The **binary tree** of ***height*** **h** with the ***minimum*** **number of nodes** is a tree where **each node** has ***one*** **child**:

    - Because the ***height*** **= h**, the are ***h*** **edges**

    - ***h*** **edges** connects ***h+1*** **nodes**

      - Therefore, the ***minimum*** **number of nodes** in a **binary tree** of ***height*** **h** = ***h*** **+ 1**

- **Fact:**

  - The ***maximum* number of nodes** in a **binary tree** of ***height* h** = $2^{h+1} - 1$

  Proof:

  - The ***perfect* binary tree** has the **maximum number of nodes**

  - We have already shown that:
    - # nodes in a ***perfect* binary tree** = $2^{h+1} - 1$