Data Structures

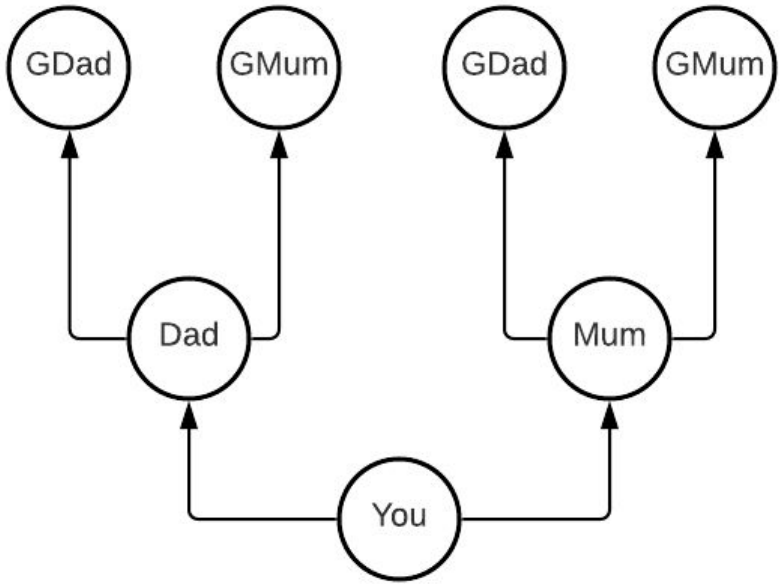
Trees

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

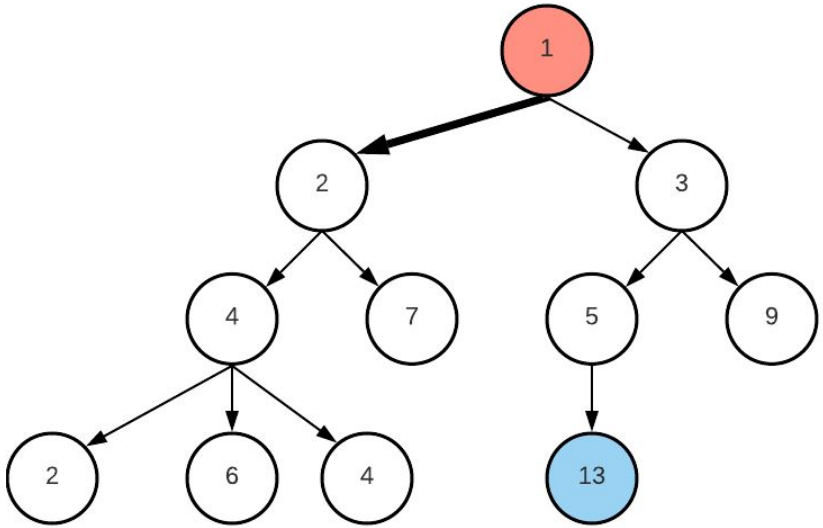
Your family tree

- Have you ever drawn up your big family tree?
- We know a tree has
  - a **Root**: **You** in this case
  - Branches (**edges**) such as You⇒Dad
  - **Leaves** such as GrandDad
- How can we represent such information in computer?
  - Imagine a deep tree!
  - Recall stack/queue/linked list are **linear**



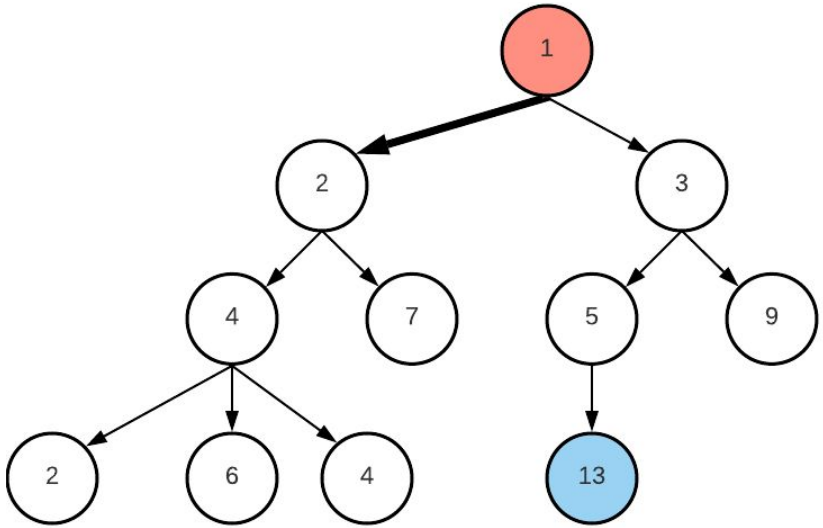
Tree Data Structure

- The tree data structure is used to represent **trees-like information**
  - The tree is usually **upside-down**
- Each circle is called a node (or vertex)
  - Node may has values (numbers, letters, strings, objects, whatever)
  - Node with value (1) is called **root**
  - Node(1) has 2 children: node(2) and node(3)
  - Node(4) has 3 children
  - Node(13) has no children. We call it **leaf**
- The link between 2 nodes is **edge**
  - Sometime an edge have a value
  - E.g. road length between 2 cities



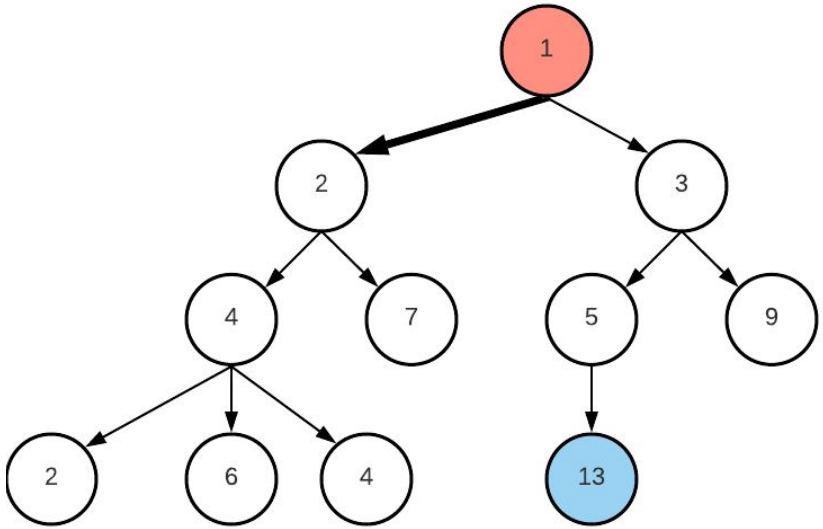
Tree Data Structure: Relations & Levels

- Node(1) has 2 **children**: 2 and 3
- The **parent** of Node(7) is node(2)
- Nodes {5, 9} are siblings (brothers)
  - Same for {2, 6, 4} with common parent 4
- This tree has 4 levels:
  - **Level 0** has nodes: 1
  - Level 1 has nodes: 2, 3
  - Level 2 has nodes: 4, 7, 5, 9
  - Level 3 has nodes: 2, 6, 4, 13



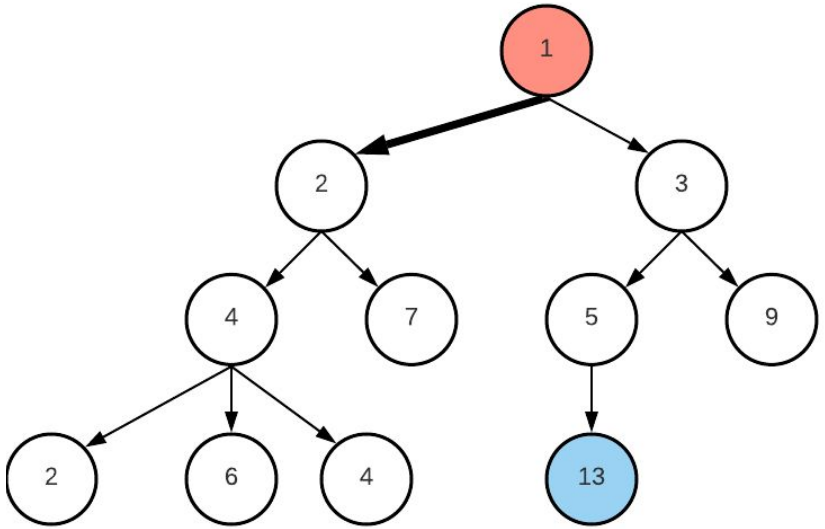
Tree Data Structure: Height

- Each tree has height: the **number of edges** on the **longest** downward path between the root and a **leaf**
  - Height of node (1) = 3
    - Longest path is: 1 ⇒ 2 ⇒ 4 ⇒ 6
  - Height of node (3) = 2
  - Height of node (4) = 1
  - Height of node (13) = 0
- Tree of N levels has N-1 height
  - We refer to it as h (**height of root**)



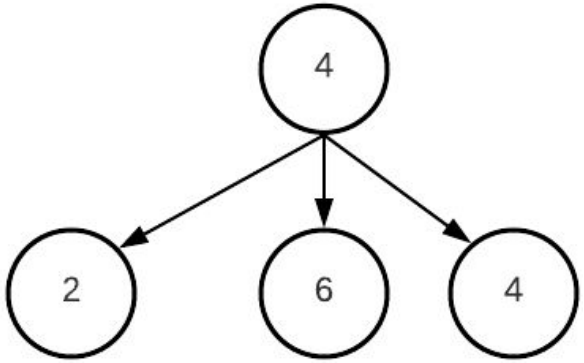
Tree Data Structure: Depth

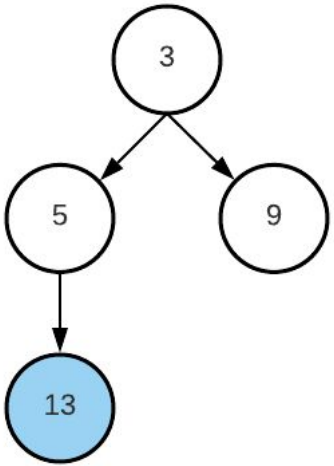
- Node's Depth = the number of edges from the node **to the root node**.
  - Depth(root) = 0
  - Depth(4) = 2: 1 ⇒ 2 ⇒ 4 [2 edges]
  - Depth(6) = 3: 1 ⇒ 2 ⇒ 4 ⇒ 6 [3 edges]
  - Height(6) = 0
- So depth is about going to up
- But height is about going to down

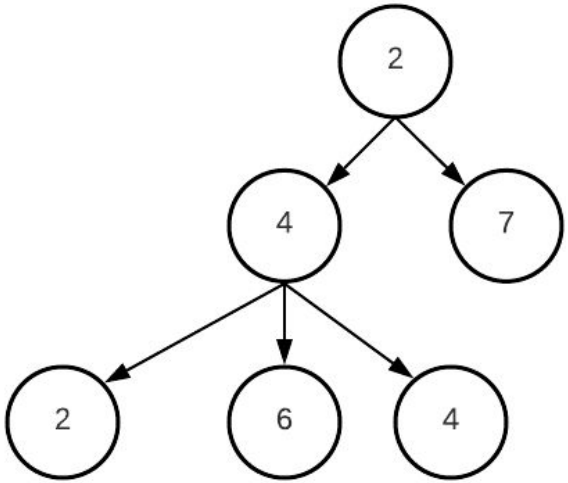


Tree Data Structure: Subtrees

- Each node with **all its nodes** below it are called a subtree: e.g. 2, 3, 4
- This is a recursive nature. That is why **recursion** is very common with trees

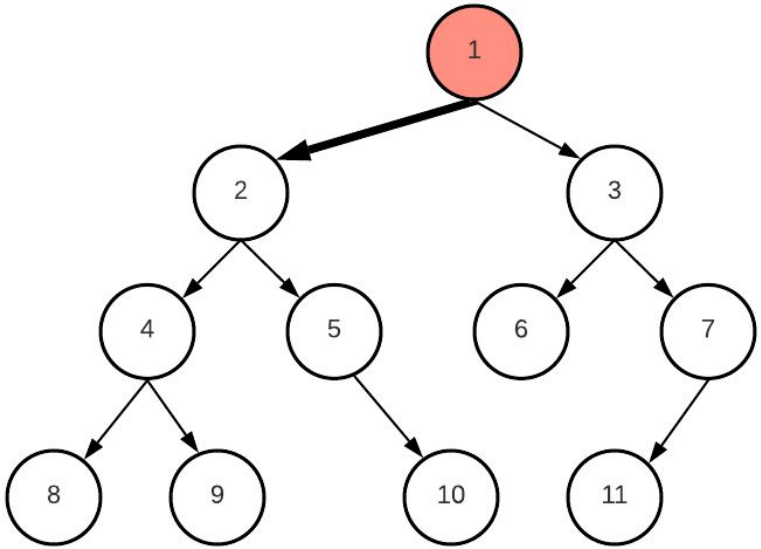






Binary Tree

- A tree in which each node has at **most two children: left** and **right** nodes
- **Left** of node(1) is node(2)
  - **Right** of node(1) is node(3)
- Left of node(4) is node(8)
  - Right of node(4) is node(9)
- Right of node(5) is node(10)
  - But it **doesn't** have a **left** node
- Left of node(7) is node(11)
  - But it **doesn't** have a **right** node
- Node (10) is **leaf** ⇒ **No** children



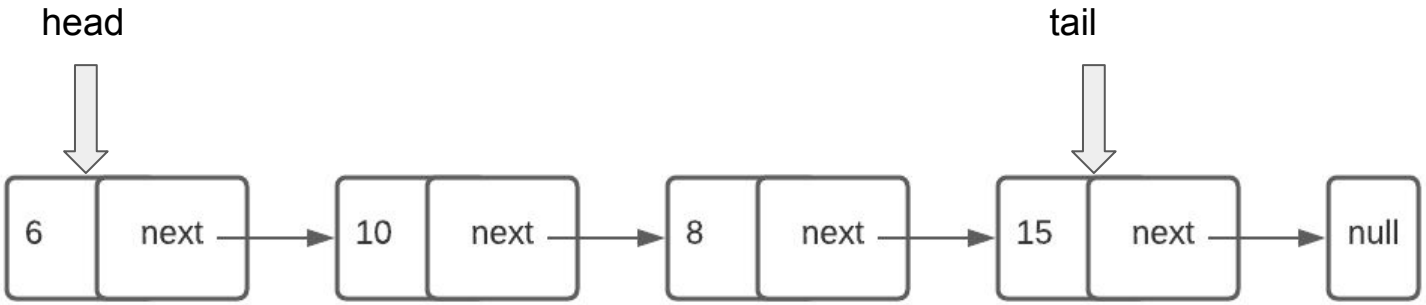
Binary Tree types

- There are many types of **binary trees**. They share some properties, but also each has its own design or characteristics
  - We will study:
  - Binary **Search** tree
  - **Balanced** Trees (AVL)
- **Trie**: General Tree

- AA tree
- AVL tree
- Binary search tree
- Binary tree
- Cartesian tree
- Conc-tree list
- Left-child right-sibling binary tree
- Order statistic tree
- Pagoda
- Randomized binary search tree
- Red–black tree
- Rope
- Scapegoat tree
- Self-balancing binary search tree
- Splay tree
- T-tree
- Tango tree
- Threaded binary tree
- Top tree
- Treap
- WAVL tree
- Weight-balanced tree

From linked list to binary tree

- You can think of a linked list a **special case** of binary tree
  - Each node has a single child only (->next), except the last has nothing (->next = null)
  - **Head** is the **root** node and tail is a leaf node!
- To code a binary tree, we simply extend it to have 2 children (2 next)



Your turn

- Similar to linkedlist, we need
  - Struct for node content
  - The BinaryTree class itself
- Try to design their attributes!
  - Follow linked style

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."