

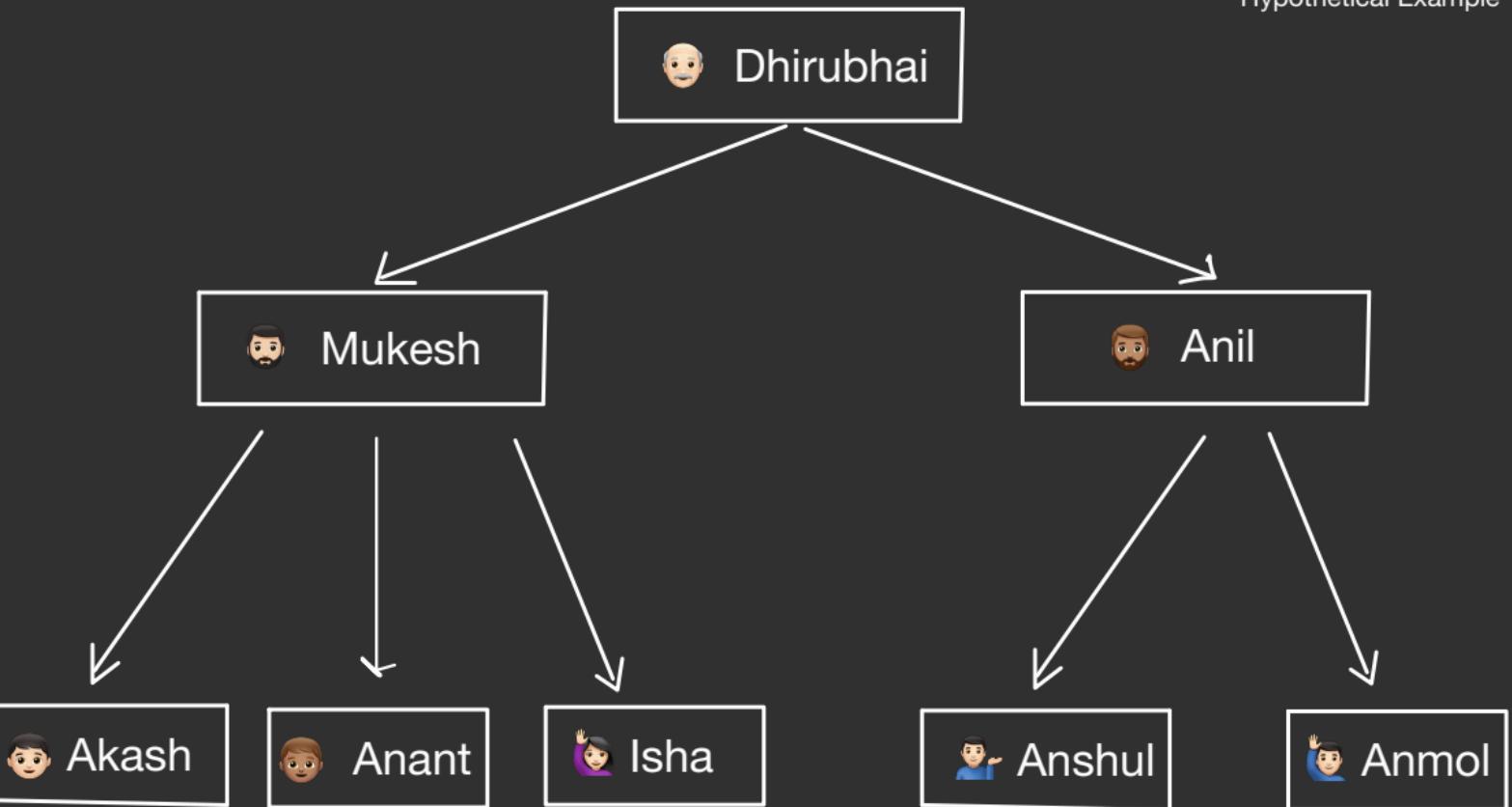
## L74

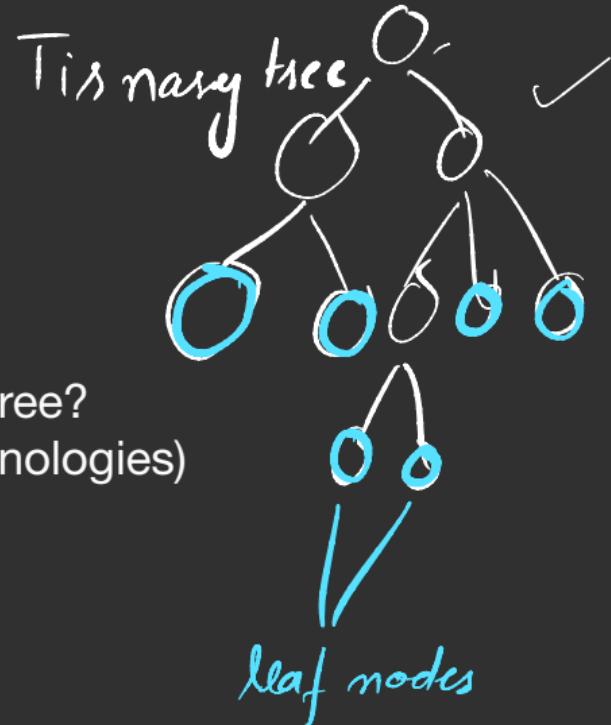
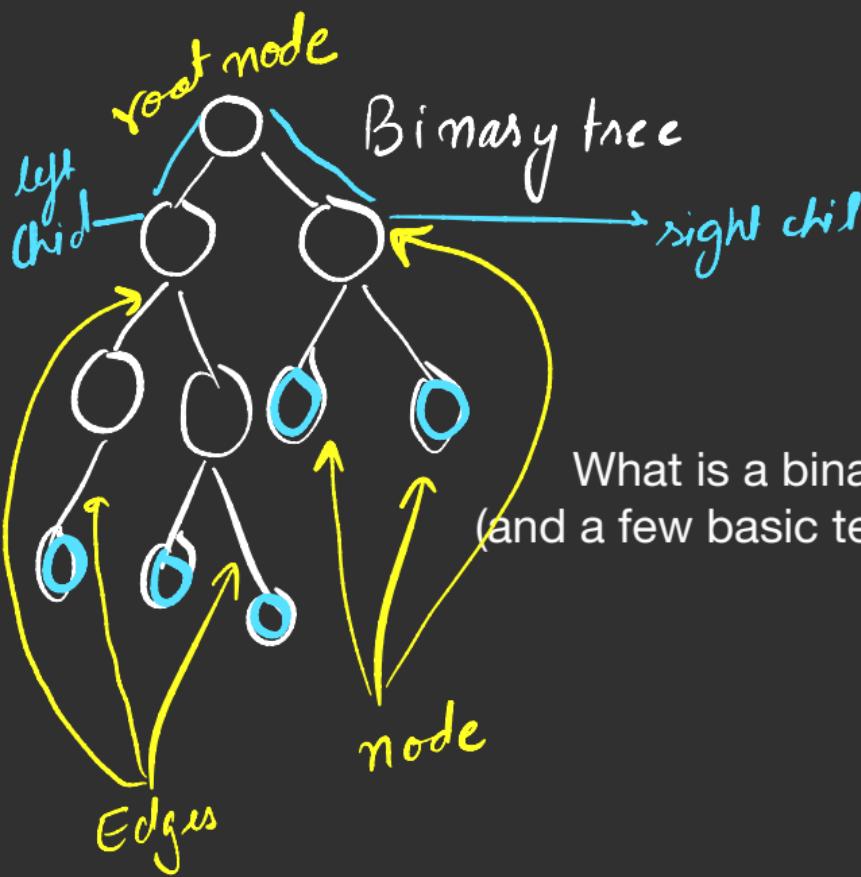
# Binary Trees : Introduction

Join Discord - <https://bit.ly/ly-discord>

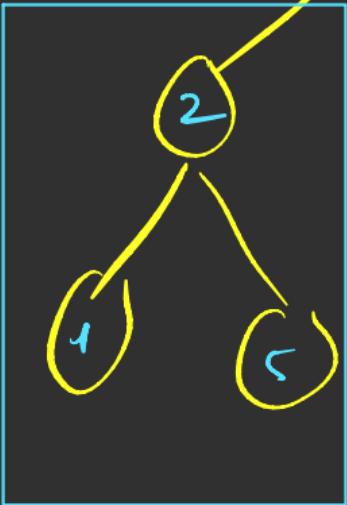
# Recap

1. A couple of DP with Probability Problems
2. Pulkit knows the specifics xD

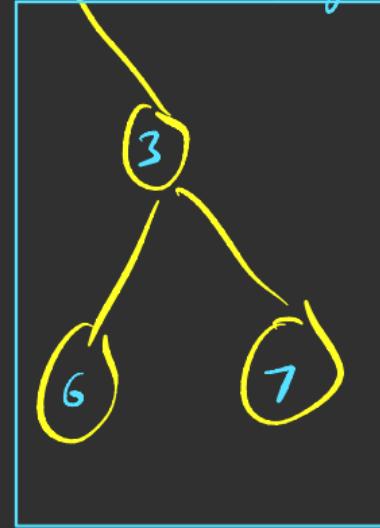




Left subtree



Right subtree



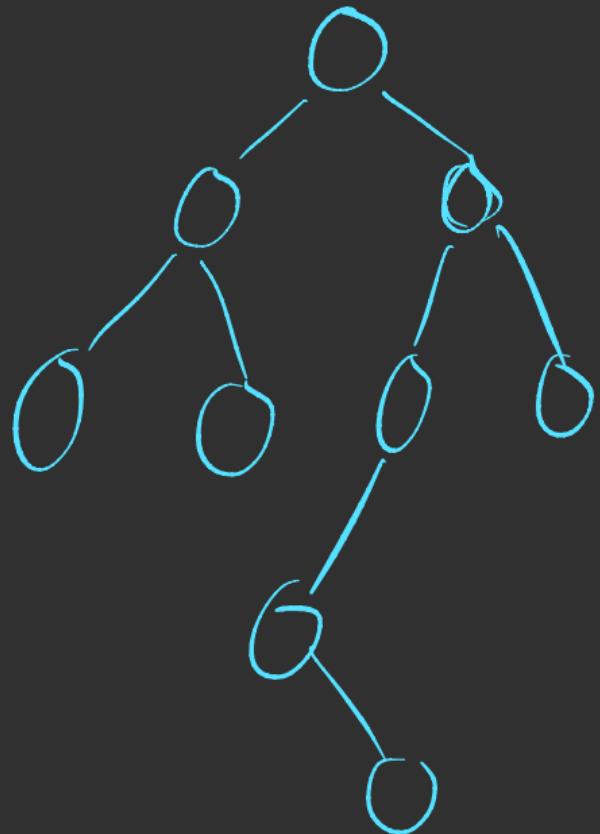
Height

3 node

2 edge

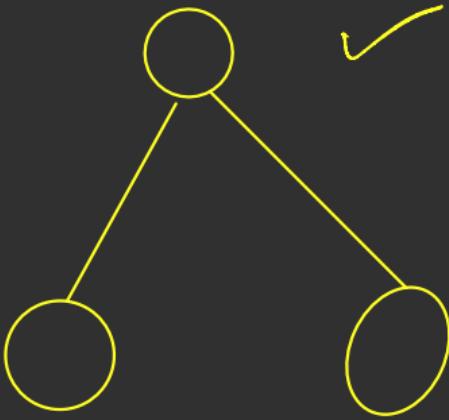
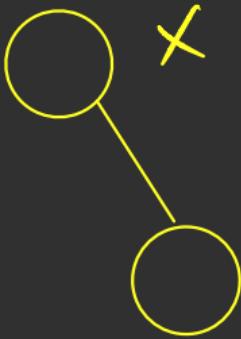
Ancestors of 5 → 2, 1

Descendents of 1 → 2, 3, 4, 5, 6, 7



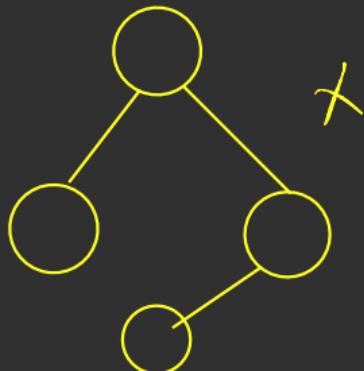
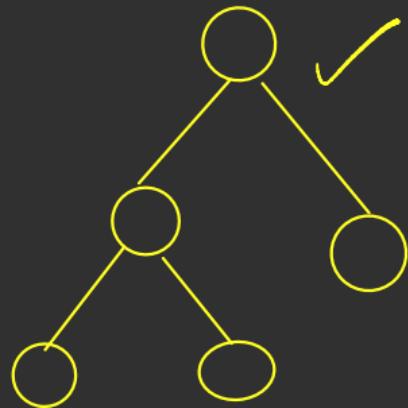
Height  
Nodes 5  
Edges 4

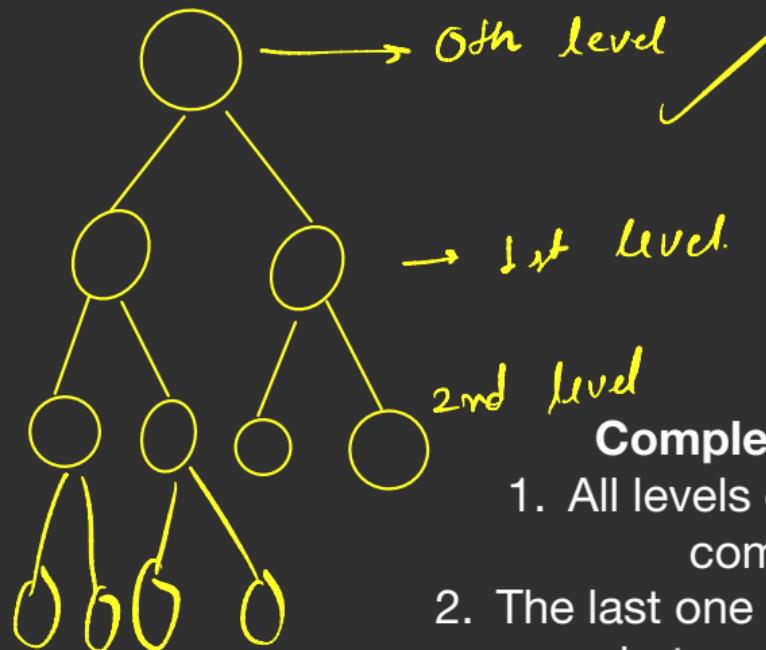
Let's look at a few types of binary trees



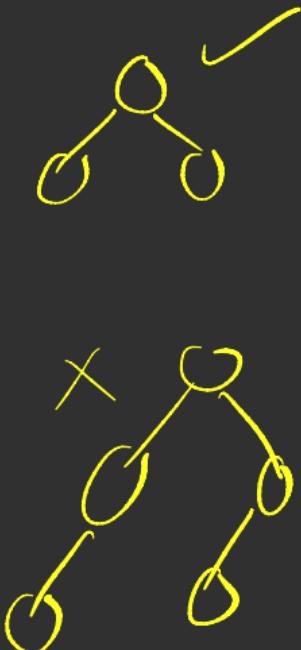
## Full Binary Tree

Every node has either 2 children or no child at all



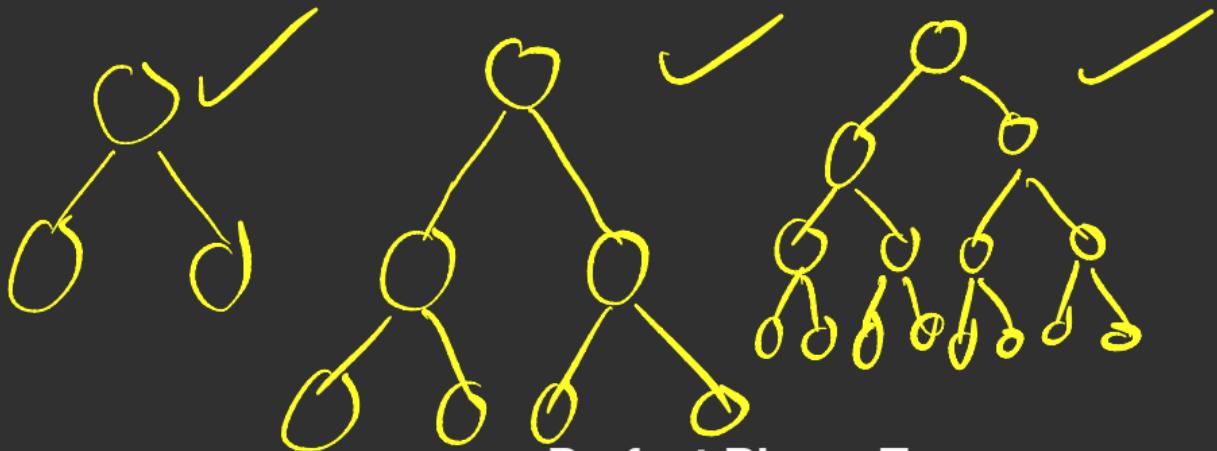


$n^{\text{th}}$  level can have  $2^n$  nodes now



### Complete Binary Tree

1. All levels except the last one completely filled.
2. The last one partially/fully filled, but whatever is filled is a prefix

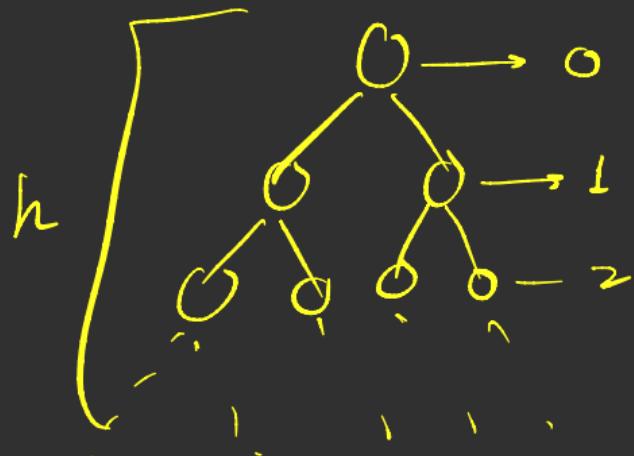


### Perfect Binary Tree

A complete binary tree that has last level also completely filled. In other words, all leaf nodes should be on the same level.



Calculate number of nodes in a perfect BT  
if height (node) is  $h$



0

$$2^0 + 2^1 + 2^2 + 2^3 \dots \dots \dots$$

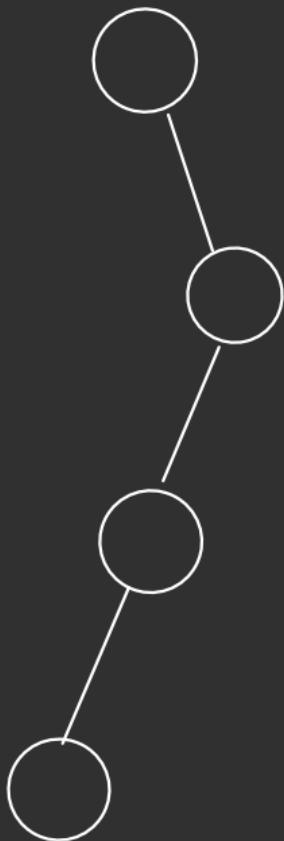
$$1 + 2 + 4 + 8 \dots \dots \dots$$

$$a \left[ \frac{y^h - 1}{y - 1} \right]$$

$$2^{h-1}$$

$$\boxed{2^h - 1}$$





## Degenerate Binary Tree

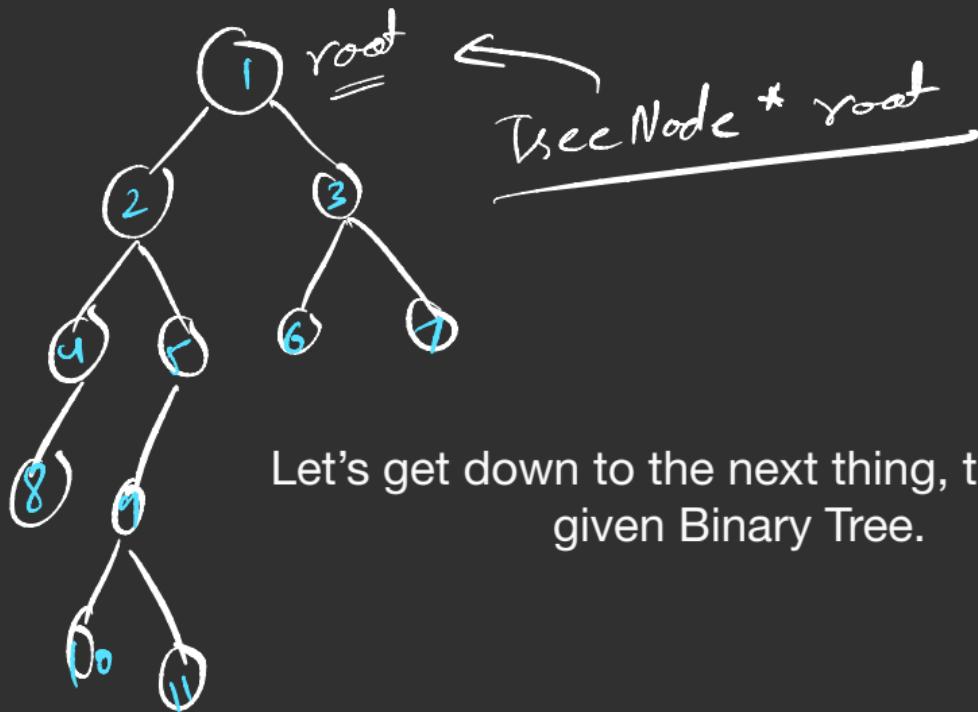
Every node has either 1 child or no children at all

```
ListNode {  
    int data;  
    ListNode* next;  
}
```

```
TreeNode <  
    int data;  
    TreeNode* left;  
    TreeNode* right;  
}
```

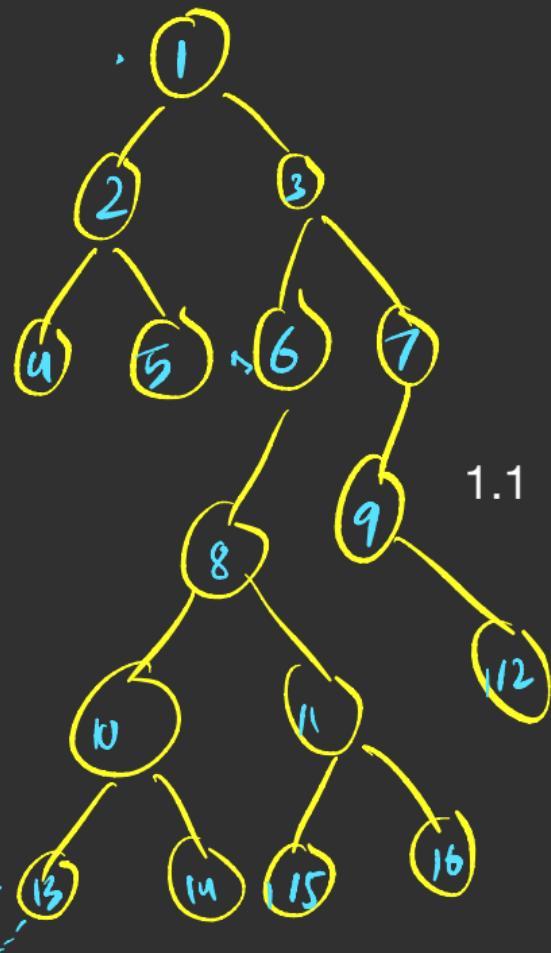
How to store different nodes?

Taking input & creating a BT?



Let's get down to the next thing, traversing a given Binary Tree.

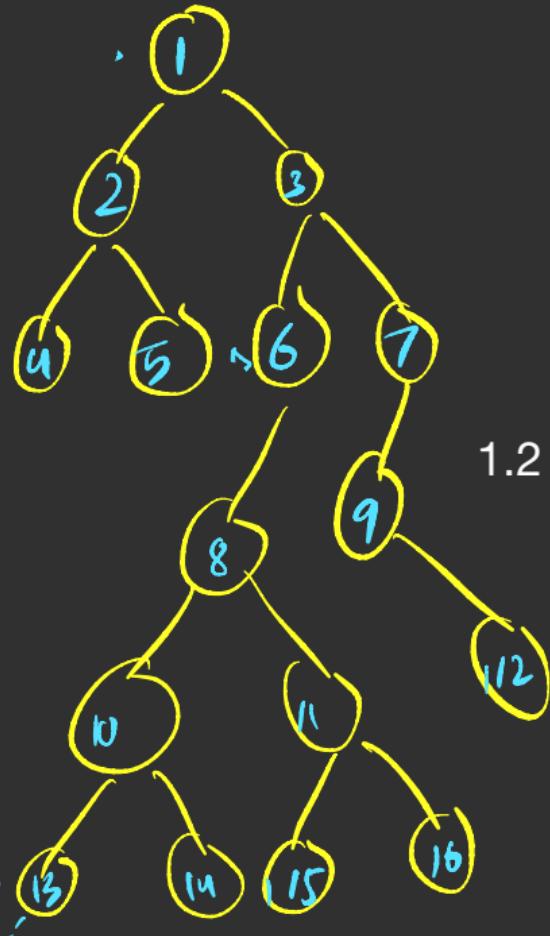
## 1. Depth First Search



### 1.1 In-Order Traversal

```
inorder(1)
{
    if (root == null) return;
    inorder(2); left
    print(1)
    inorder(3); right;
}
```

4 2 5 1 13 10 14 8 15  
11 16 6 3 9 12 7

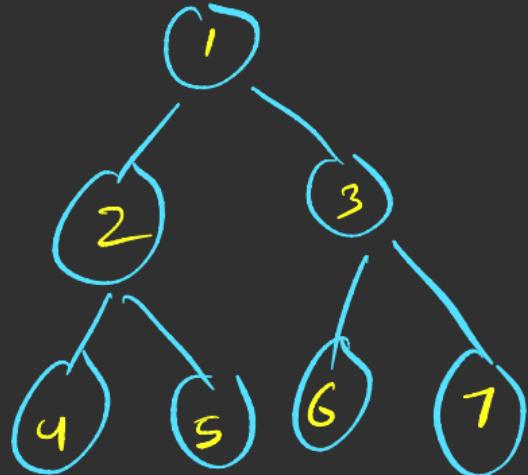


1 2 4 5 3 6 8 10 13 14  
11 15 16 7 9 12

## 1.2 Pre-Order Traversal

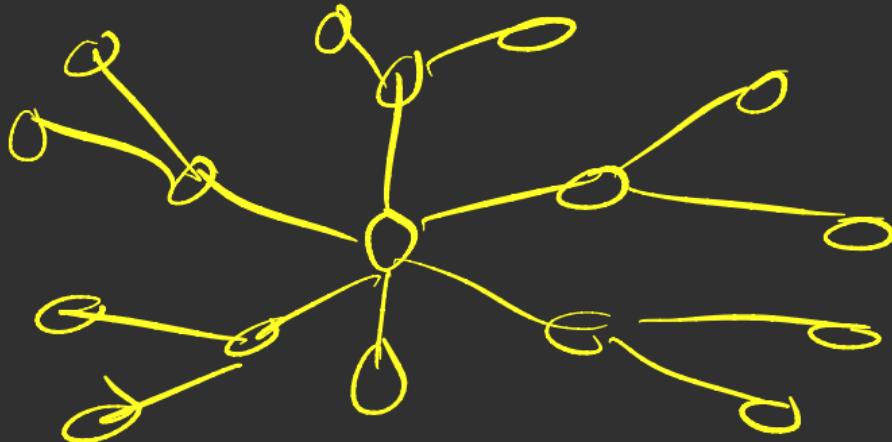
Time  $O(n)$   
Space  $O(h)$

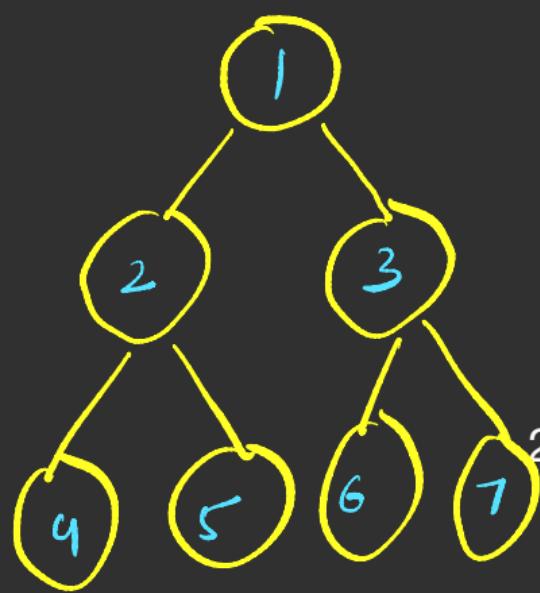
## 1.3 Post-Order Traversal (HW)



1 2 3    4 5 6 7  
— —

## 2. Breadth first Search





3 4 5

## 2.1 Level Order Traversal

Without a queue?

# Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!