DSA SERIES

- Learn Coding

Topic to be Covered today

Binary Tree

LETS START TODAY’S LECTURE

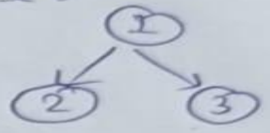
Lecture - 50

Properties of Binary Trees

* **Node** : • The building block of tree.
- It contains data and links to left and right children.

* **Root** : The topmost node of a tree.

- Example: In this tree, 1 is root.



* **Parent and child** :
- A node that has links is called a parent.
- The connected nodes below it are children.
- Example :- 1 is ~~is~~ the parent of 2 and 3.

* ~~Leaf Node~~ :
- ~~A node that links is called a parent.~~
- The connected nodes

* **Leaf Node**:
- A node with no children.

Example :- 2 and 3 are leaf nodes.

* Height of a node

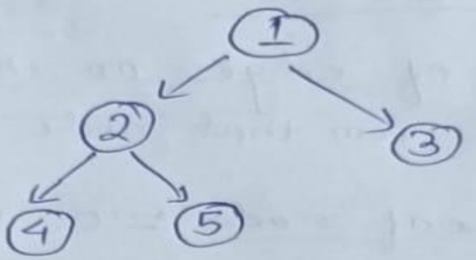
↳ The number of edges on the longest path from that node down to a leaf.

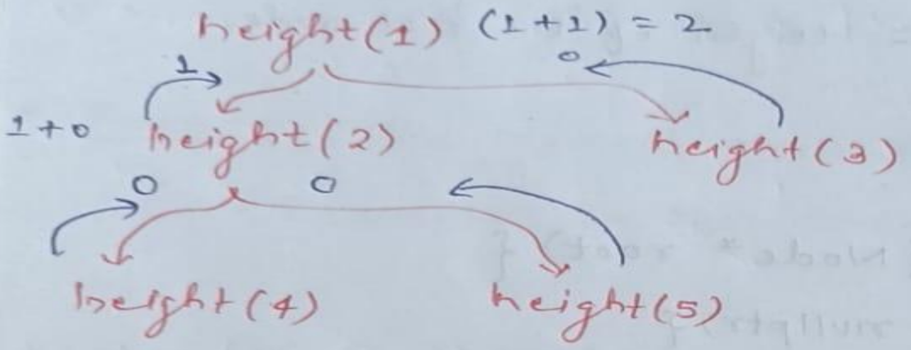
- **Height of a leaf node** = 0 (since there are no edges below it).
- **Height of a Tree** = Height of the root node.

Code:-

```
int height (Node* root) {
    if ( root == nullptr) {
        return -1;
    }
  int leftHeight = height (root -> left);
  int rightHeight = height (root -> right);

  return 1 + max(leftHeight + rightHeight);
}
```

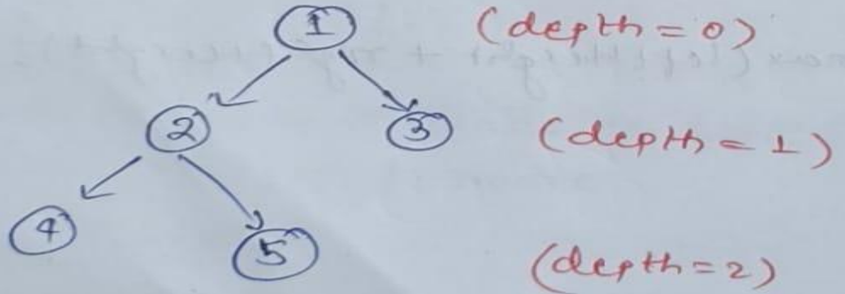
Example





Depth of Node

↳ The depth of a node is the distance (number of edges) from the root node to that particular node.



Code.

```
int depth (Node* root, int target){
  if (root == NULL) return -1;
  if (root -> data) == target)  return 0;
 int leftDepth = depth(root -> left, target);
  if (leftDepth != -1) return leftDepth+1;

 int rightDepth = depth(root -> right, target);
 if (rightDepth != -1) return rightDepth+1;
 return -1;
  }
```

* Level of a Node

=> Depth +1

* Size of a tree

Total number of nodes in the tree

* Degree Of a Node

The number of children a node has (0, 1, or 2)

* Maximum nodes at level l =

$= 2^l$

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

* Maximum nodes in a tree of height h:

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

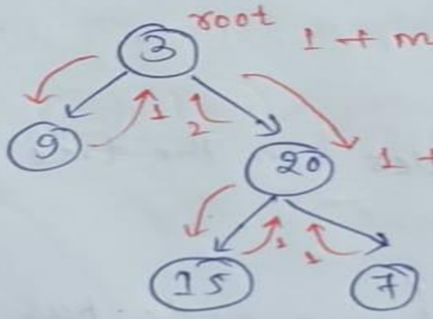
$h = 3, \quad 2^4 - 1 = 8 - 1$

$= 7$

Leetcode Question - 111

Minimum Depth of Binary Tree

Using DFS



$1 + \min(L, R) \Rightarrow 1 + \min(1, 2)$

$= 1 + 1$

$= 2$

$1 + \min(1, 1)$

Using BFS

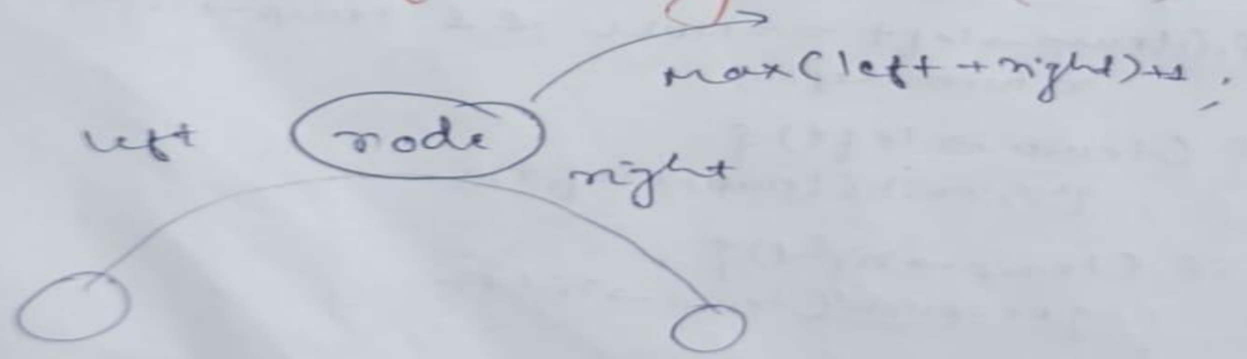
Here we will process level by level.

```
int minDepth(TreeNode* root){
   if (root == NULL){
        return 0;
    queue <TreeNode *> que;
    que.push(root);
    int depth = 1;
  while (!que.empty()){
     int n= que.size();
     while (n--){
    TreeNode * temp = que.Front();
      que.pop();
    if (temp->left == NULL && temp->right == NULL)
          return depth;
    if (temp -> left){
          que.push(temp -> left);

     if (temp -> right){
          que.push( temp -> right);
      }
     depth++;
   }
  return -1;
}
```

DFS code.

```
int minDepth(TreeNode *  root){
    if(!root) {
        return 0;
    }
   if (root -> left == NULL && root -> right == NULL)
        return 1;

  int L= root ->left != NULL ? minDepth (root -> left)
                                  :INT_MAX ;
 int R = root -> right != NULL ? minDepth (root -> right):
                                     INT_MAX;
 return 1 + min(L, R);
 }
```

* Diameter of Binary Tree (543)



```
int diameter(TreeNode* root){
    if (root == NULL){
        return 0;
    }
  int result = INT_MIN;
  solve(root, result);
  return result;
 }

int solve(TreeNode* root, int & result){
   if (root == NULL) return 0;
  int left = solve (root->left, result);
  int right = solve (root->right, result);
  result = max(result, left + right);
   return max(left, right) + 1;
}
```

Learn coding

Thank You