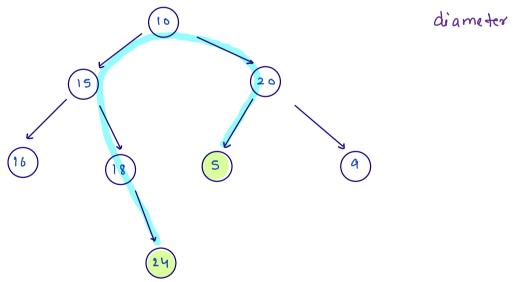
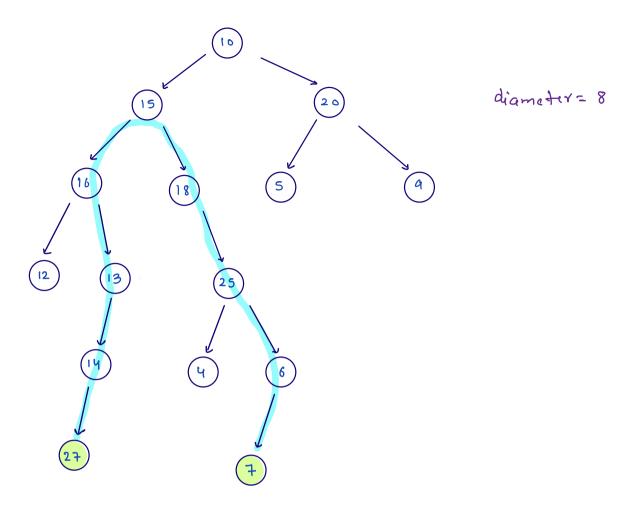
Agenda

- 1) Diameter of Binary tree
- 2) Serialize & Deserialize Binary tree
- 3) Treemap gntroduction & usage

Q-1 Given root of a binary tree, find its diameter. Note: The diameter of binary tree is the length of the longest poth blw any two nodes. This poth may or may not pass through the root.



diameter = 5



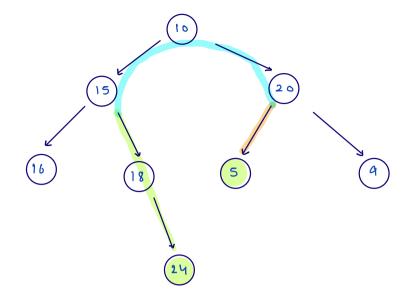
Un = distance from

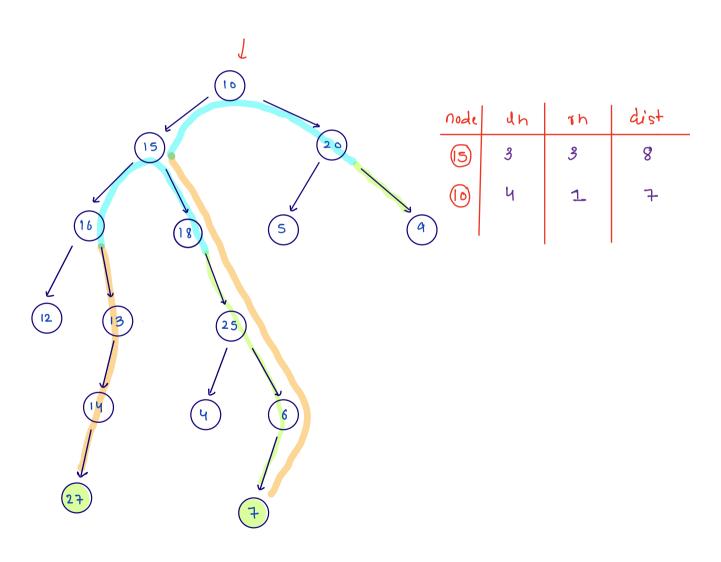
dest child to deepest

node in lest subtree

sh= distance Jrom sight child to deepert node in sight subtree

dist = 1h+rh+2





```
int max Dist = 0;
      height (Node 100t) ?
 int
      ij( root = = nul) {
            idum -1)
       3
      Int un= height (800 t. left)
       int the beight (root right);
       int mh = Math. max (dh, 8h)+1;
        int d = dh+ rh+2;
        if (d> max Dist) {
             max Dist = d;
         return mh;
3
     diameter ( Node Trot) }
†ní
      max sist = 0;
       height (root);
        odyra max Dist;
 3
```

```
mh = 3
int max Dist = 0;
                                                             d = 5
                                                         10
                                                                    1
int height (Node 100t) }
                                                                          mh = 1
   il ( root = = null) {
                                                   mh=2
d=3
                                               15
                                                                           d = 2
         return -1;
    3
    Int un= height (800 t. left)
                                                        mh=1
                                       (16
                                                    18
                                                        d=1
    int the beight (root right);
                                           d=0
    int mh = Math. max (dh, 8h) +1;
     int d = dh+ th+2;
      if (d> max Dist) {
          max Dist = d;
                                                    maxDist = 8285
      return mh;
3
       diameter ( Node Trot) }
 int
        max sist = 0;
        height (root);
```

odyrn max Dist;

3

```
mn = 4
in+ max Dist = 0;
                                                           10
                                                                  d=5
int height (Node 100t) }
                                                      mh = 3
                                                  15
   il( root = = nul) {
                                                       d = 6
         ictum -1;
    3
                                              hh=2
                                                          mh=2
   Int un= height (800 to dest)
                                          (16
                                              d=2
                                                           d = 2
    int the beight (took right);
    int mh = Math. max (lh, 8h)+1;
                                                mh=1
                                                              mh=1
                                                d=1
                                                           24
                                                               d = 2
     int d = dh+rh+2j
     if (d> max Dist) {
          max Dist = d;
                                                                  71
      rourn mh;
3
                                                       max Dist = 8 226
 int diameter ( Node root) }
     max sist = 0;
      height (root);
```

return max Dist;

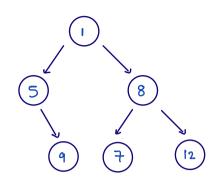
3

2 ٥

https://leetcode.com/problems/diameter-of-binary-tree/

```
class Solution {
   int maxDist = 0;
    public int diameterOfBinaryTree(TreeNode root) {
       maxDist = 0;
       height(root);
       return maxDist;
   public int height(TreeNode root) {
       if(root == null) {
           return -1;
       int lh = height(root.left);
       int rh = height(root.right);
       int mh = Math.max(lh,rh) + 1;
       int d = lh + rh + 2;
       if(d > maxDist) {
           maxDist = d;
       return mh;
```

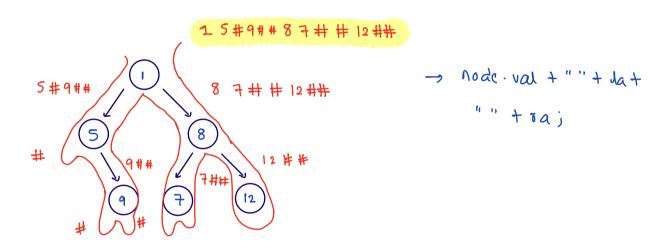
0.2 Serialize and Deserialize a Binary tree.



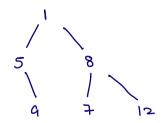
Serialize: convert into of BT into a String

Deservative: with this same string convert back to the binary tree

Serilization: Binary tree into into string.



1 5#9##8 7# # 12##



```
String serialization (Node root) {

ij(root == n wh) {

roturn "#";

string da = Serialization (root. dept);

String ra = Serialization (root. right);

String ma = node. val + "" + da + "" + ra;

roturn ma;

3
```

gan em

```
String serialization (Node *00t) {

ij(root == n wh) {

toturn "#";

string wa = Serialization (*00t. left);

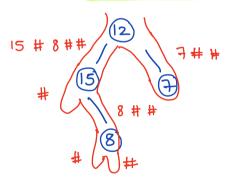
String ra = Serialization (*00t. right);

String ma = node.val + "" + Ja + "" + ra;

roturn ma;

3
```

12 15 # 8 # # 7 ##



Deservation: Joom the servatived string that you have created in above logic, give back the tree.

L 12 15 # 8 # # 7 # #

[i] I see a data - node creation

- left call

- right call

Plse Tolurn null

```
or serialized string
int idx = 0;
        Describlize (String Str) }
Node
      5tring [] abs = Sto. split (" ");
      idx = 0;
      return haper (ass);
3
Node holper (String [] arr) ?
      if (arr [idx]. equals ("#")) {
            idx++;
            return null;
      3
       else 1
            int val = Integer. parse Int (arridx))
            Node nn= now Node (val);
            idx++;
            nn. Lyt = helper (arr);
             nn. right = helper (arr);
             return node;
       3
3
```

```
Node haper (String [] arr) ?
                                                                                   Ĺ
  ij (arr [idx]. equals ("#")) { arr: ["12","|5","
         idx++;
        return null;
                                                                         null
   3
                                                                         AUCT
   else 1
                                                       (12)
        int val = Integer. ParseInt (arridx))
                                                                         معللاو
        Node nn= now Node (val);
                                                  (3)
                                                                        JULL
         idx++;
                                                                         8 X X
         nn. Lest = helper(arr);
                                                             nue nue
                                             null
                                                                        Auth
         nn. right = helper (arr);
         THUEN ON ;
                                                      nue nue
   3
3
```

```
public class Codec {
   // Encodes a tree to a single string.
   public String serialize(TreeNode root) {
       if(root == null) {
           return "#";
       String la = serialize(root.left);
       String ra = serialize(root.right);
       String ma = root.val + " " + la + " " + ra;
       return ma;
   }
   // Decodes your encoded data to tree.
   public TreeNode deserialize(String data) {
       String[]arr = data.split(" ");
       idx = 0;
       return helper(arr);
   }
   int idx = 0;
   public TreeNode helper(String[]arr) {
       if(arr[idx].equals("#")) {
           idx++;
           return null;
       }
       else {
           int val = Integer.parseInt(arr[idx]);
           TreeNode nn = new TreeNode(val);
           idx++;
           nn.left = helper(arr);
           nn.right = helper(arr);
           return nn;
```

Introduction to TreeMap

Treemap is a sorted Hashmap.

Ly based on key values

TreeMap < String, Integer? tm = New Tree Map <>();

tm. put ("India", 244);

tm. put ("Australia", 444);

tm. put ("(hina", 434);

tm. put ("Bharat", 416);

tm. put ("Pak", 384);

sorin (tm. contains key ("Eng")); — Jalse

sorin (tm. qet ("Bharat")); — 416

nash Map	TreeMap
lut, containskey, got,	Put, containskey, get,
remove	remove, culting key, floorkey
0(1)	0(109 n)

Ide link:

https://www.interviewbit.com/snippet/9bd6263c3cdcbe54e0fd/