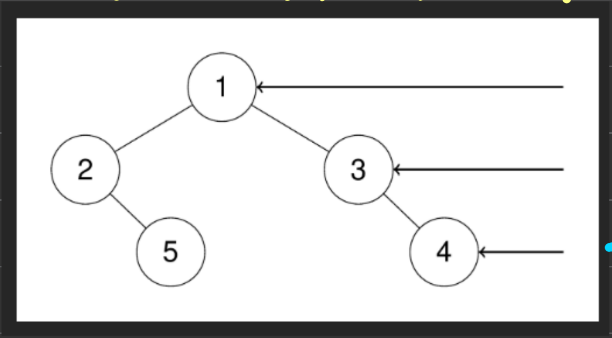
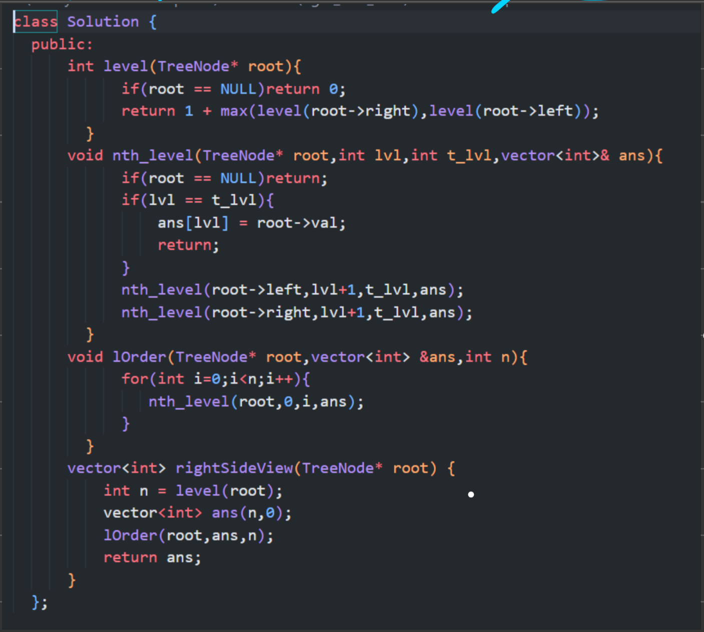
APPROACH : 01



make A vector of size same as size of levels of tree...

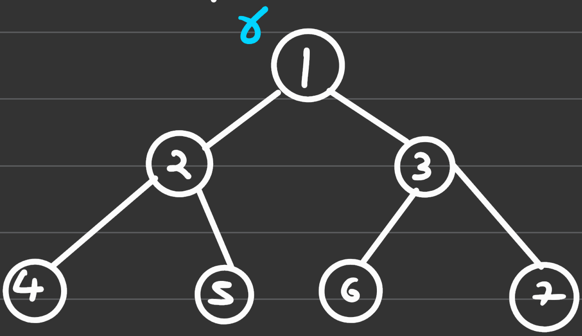
HAR Level per JANA Aur jaise element ko starting se print se push karte jao value khud update hoke right most element Banjayega...

Method : 1) Coding Implementation...



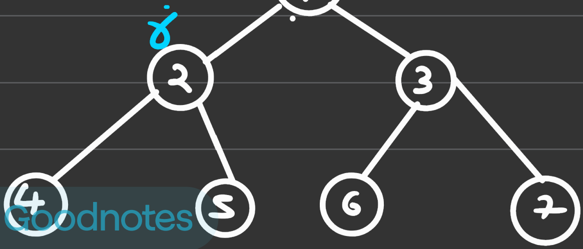
```
class Solution {
  public:
    int level(TreeNode* root){
          if(root == NULL)return 0;
          return 1 + max(level(root->right),level(root->left));
      }
    void nth_level(TreeNode* root,int lvl,int t_lvl,vector<int>& ans){
          if(root == NULL)return;
          if(lvl == t_lvl){
              ans[lvl] = root->val;
              return;
          }
          nth_level(root->left,lvl+1,t_lvl,ans);
          nth_level(root->right,lvl+1,t_lvl,ans);
      }
    void lOrder(TreeNode* root,vector<int> &ans,int n){
          for(int i=0;i<n;i++){
             nth_level(root,0,i,ans);
          }
      }
    vector<int> rightSideView(TreeNode* root) {
        int n = level(root);
        vector<int> ans(n,0);
        lOrder(root,ans,n);
        return ans;
    }
};
```

=> As simple as ki har level per jao And Array k same index per Values update karo..



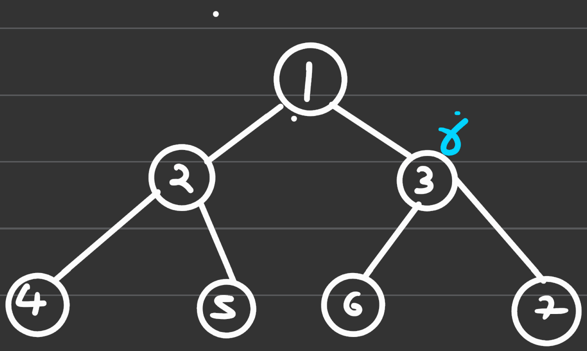
0	1	2
1		

t_lvl=0



0	1	2
1	2	

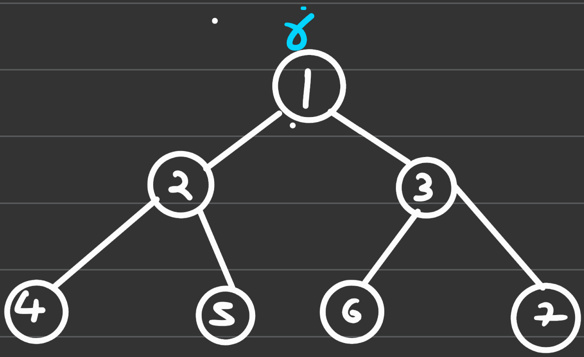
t_lvl=1



0	1	2
1	~~2~~3	

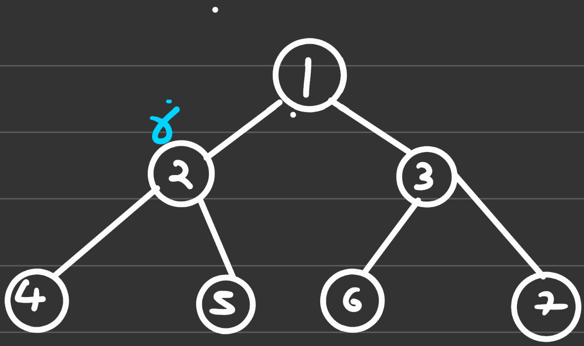
t_lvl =1

tlvl=2



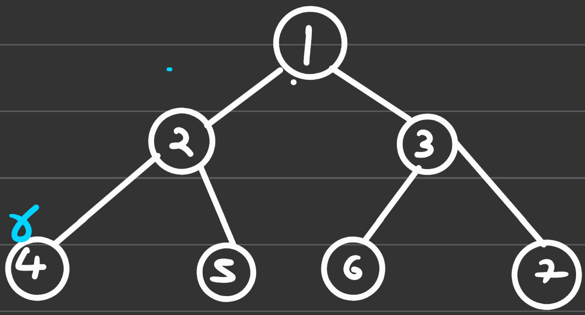
0	1	2
1	~~2~~3	

t_lvl =2



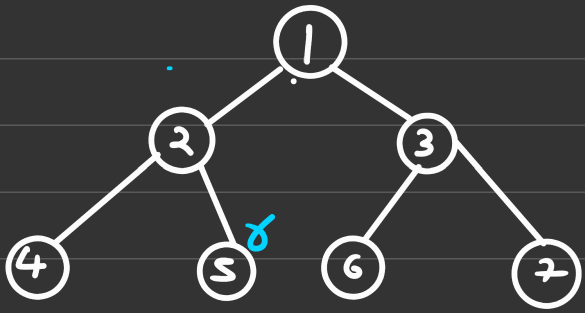
0	1	2
1	~~2~~3	

t_lvl =2



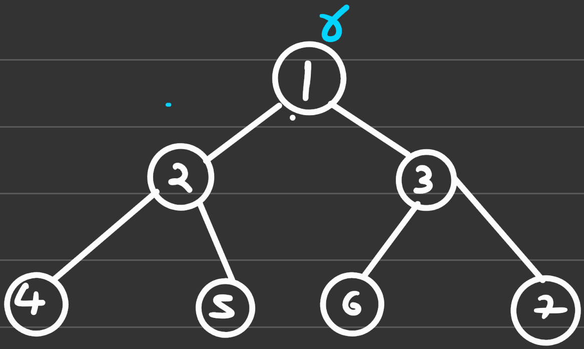
0	1	2
1	~~2~~3	4

t_lvl =2



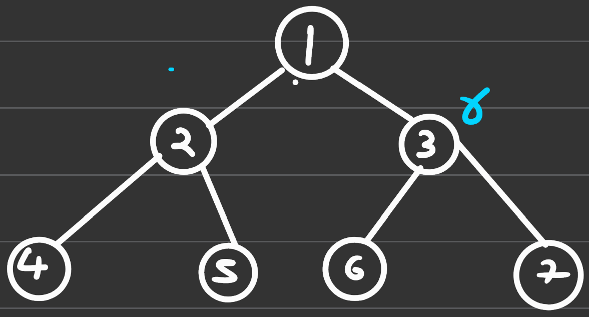
0	1	2
1	~~2~~3	~~4~~5

t_lvl =2



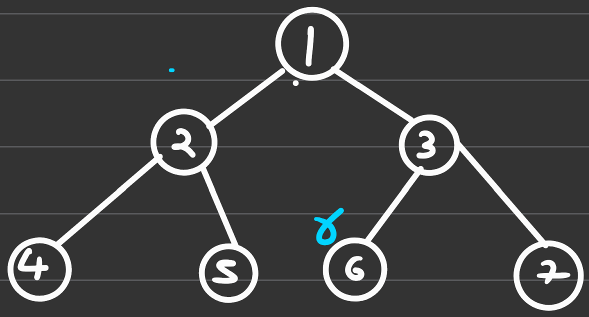
0	1	2
1	~~2~~3	~~4~~5

t_lvl =2



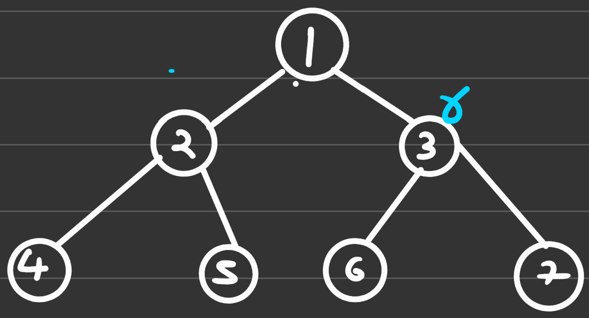
0	1	2
1	~~2~~3	~~4~~5

t_lvl =2



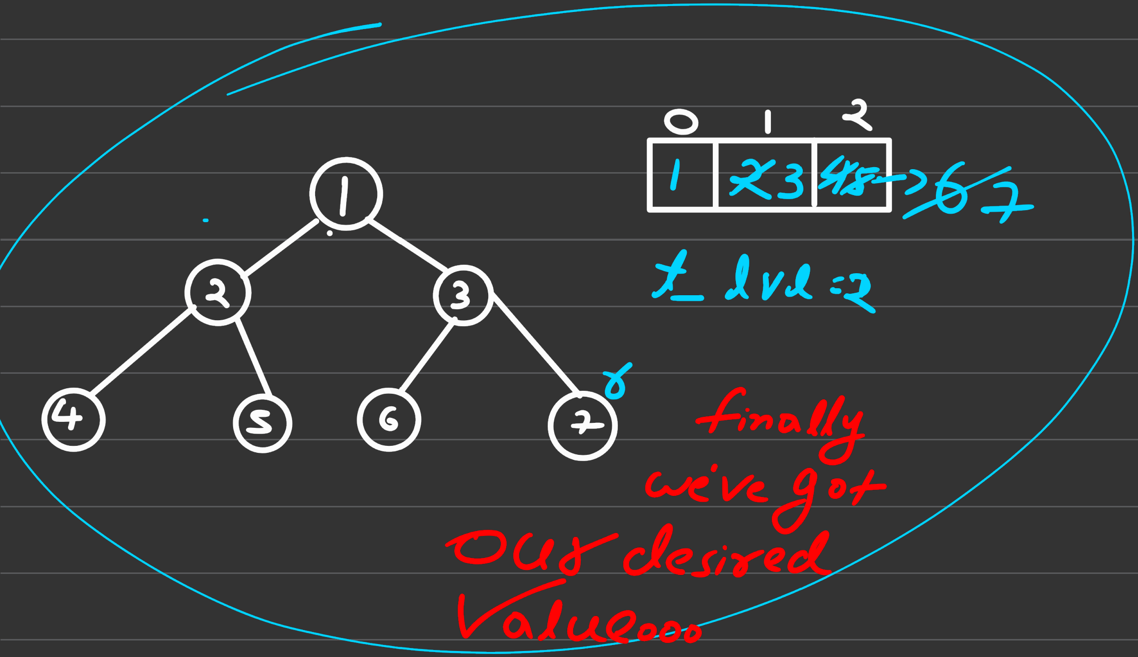
0	1	2
1	~~2~~3	~~45~~->6

t_lvl =2



0	1	2
1	~~2~~3	~~45~~->6

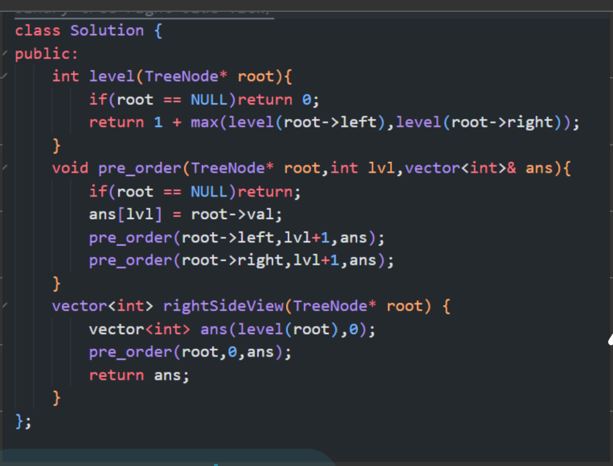
t_lvl =2



finally we've got our desired valueooo

Method:2 using preorder traversal:-

↳To update Array



```
class Solution {
public:
    int level(TreeNode* root){
        if(root == NULL)return 0;
        return 1 + max(level(root->left),level(root->right));
    }
    void pre_order(TreeNode* root,int lvl,vector<int>& ans){
        if(root == NULL)return;
        ans[lvl] = root->val;
        pre_order(root->left,lvl+1,ans);
        pre_order(root->right,lvl+1,ans);
    }
    vector<int> rightSideView(TreeNode* root) {
        vector<int> ans(level(root),0);
        pre_order(root,0,ans);
        return ans;
    }
};
```

Everything is Same Bas traversal pre_order se karo Aur har ek node ko Array me update karna hai...

Revision 4 time perday sum
kar lena please!