5/9/24

lab - 2.

k^th Largest Sum in a Binary Tree

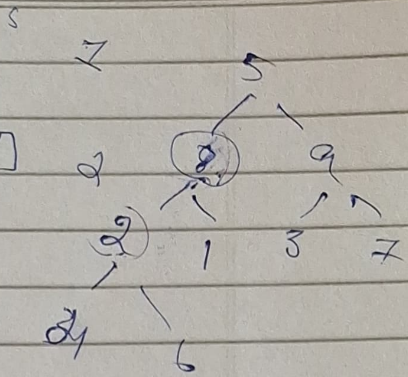
```
int height (struct TreeNode* root)
{
   if (root==NULL)
   {
     return 0;
   }
   else
   {
     int lheight =height (root->left);
     int rheight =height (root->right);
     if (lheight >rheight)
     {
       return lheight+1;
     }
     else
     {
       return rheight+1;
     }
   }
}
void dfs (struct TreeNode* root, int level, long
                 long* sums) {
  if (root==NULL)
  {
    return;
  }
  sums[level]=sums[level]+root->val;
  if (root->left)
  {
    dfs(root->left, level+1, sums);
  }
if (root -> right){
  dfs(root -> right, level+1, sums);
 }
}

Long long kthLargestLevelSum (struct TreeNode *root,
                                        int k)
{
    int h= height(root);
    if (k > h)
    {
     return -1
    }
  long long *sums =(long long*) calloc(h, sizeof(long long));
  dfs(root, 0, sums);
  for(int i=0; i<h-1; i++){
    for(int j=0; j<h-i-1; j++){
       If (sums[j]< sum[j+1])
       {
          long long temp=sums[j];
          sum[j]=sums[j+1];
          Sums[j+1]=temp;
     }
  }
}
Long Long largest=0;
largest =sum[k-1];
free(sums);
return largest;
}
```

DATE: / /
PAGE NO.: Usha Gold

Testcase 1

root = [5, 8, 9, 2, 1, 3, 7, 4, 6]
k = 2
Output = 13



Testcase 2
root = [1, 2, null, 3]
k = 1
Output = 3

9/5/24