

week 2

K^{th} largest sum in a Binary Tree

PAGE NO.

DATE:

7/05/2024

```
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 *sums) {
    if (root == NULL)
        return;
    sums[level] = sums[level] + root->data;
    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 n = height(root);
    if (k > n)
        return -1;
    long long *sums = (long long *) calloc(n,
                                             sizeof(long long));
    dfs(root, 0, sums);
}
```



```

for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
        if (sums[j] < sums[j + 1]) {
            long long temp = sums[j];
            sums[j] = sums[j + 1];
            sums[j + 1] = temp;
        }
    }
}

```

```

long long largest = 0;
largest = sums[k - 1];
free(sums);
return largest;
}

```

Output :-

① root = [5, 8, 9, 2, 1, 3, 7, 4, 6]

K = 2

Output = 13

② root = [1, 2, null, 3]

K = 1

Output = 3

~~NA~~
16/6/24