1/08/24 week2 k^{th} Largest Sum in A

Binary Tree

```
typedef struct TreeNode node;
int get_height (node * root)
{
    if (!root)  return 0;
    int l = get_height (root -> left) + 1;
    int r = get_height (root -> right) + 1;
    return l > r ? l : r;
}

void print_arr (long long *arr, int n)
{
    for (int i = 0; i < n; i++)
        printf ("%lld ", arr[i]);
    printf ("\n");
}

void swap (long long *a, long long *b)
{
    long long t = *a;
    *a = *b;
    *b = t;
}

int partition (long long *data, int l, int h)
{
    int i = l, j = l;
    long long pivot = data[h-1];
    for (j=l; j < h; j++)
        if (data[j] < pivot)
            swap (&data[i++], &data[j]);
    swap (&data[i], &data[h-1]);
    return i;
}
```

```
void quick_select (long long *data, int l, int h, int
{
    if (l+1 < h)
    {
        int p = partition (data, l, h);
        if (p == k)  return;
        else if (p < k)
            quick_select (data, p+1, h, k);
        else quick_select (data, l, p, k);
    }
}

long long kthLargestLevelSum ( struct treeNode * root, int
{
    int n = get_height (root);
    if (k > n)
        return -1;
    long long val, *data = (long long*) calloc (n, sizeof(long
    get_level_sum (root, 0, data);
    quick_select (data, 0, n, n-k)
    return data [n-k]
}
```

→ output

⇒ Case 1

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

k = 2

output: 13

expected: 13

Date
Page

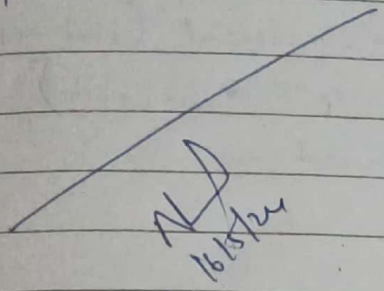
e) case 2

root = [1, 2, null, 3]

k = 1

output 3

Expected 3



16/5/24