

Office Hour #4

```
1
2 struct Node {
3     int first;
4     int second;
5     int val;
6 };
7
8 int findLargest(Node *head, int curr) {
9
10     if (head[curr].first == -1 && head[curr].second == -1)    // BASE CASE: No more
    children in tree
11         return head[curr].value;
12
13     int one, two;
14     if (head[curr].f != -1)
15         one = head[curr].first;    // get index of FIRST branch
16     else
17         one = head[curr].val;
18     if (head[curr].f != -1)
19         two = head[curr].second;    // get index of SECOND branch
20     else
21         two = head[curr].value;
22
23     return
24     (max(head[0].value,
25     findLargest(head + one - curr, one),
26     findLargest(head + two - curr, two)));
27 }    // MAY NOT BE RIGHT

1 // MODIFY FUNCTION TO RETURN BOTH THE MAX ITEM AS WELL AS ITS LEVEL, TOP LEVEL BEING 0
2 int find
```