29-2-24

Hackerrank

→ Swap Node [Algo]

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int id;
    int depth;
    struct node *left, *right;
};

void inorder (struct node* tree) {
    if(tree == NULL)
        return;
    inorder (tree → left);
    printf("%d", tree → id );
    inorder ((tree → right));
}
int main (void)
{
    int num, i=0;
    int l, r, max_depth, k;
    struct node *temp = NULL;
    scanf ("%d", &num);
    struct node *tree = (struct node*) calloc (num, sizeof (struct
    tree[0].depth = 1;
    while (i<num) {
        tree[i].id = i+1;
        scanf("%d %d", &l, &r);
        if(l==-1)
            tree[i].left = NULL;
        else {
            tree[i].left = &tree[l-1];
            tree[i].left → depth = tree[i].depth +1;
            max_depth = tree[i].left → depth;
        }
```

```
        if (r == -1)
            tree[i].right = NULL;
        else {
            tree[i].right = &tree[r-1];
            tree[i].right -> depth = tree[i].depth + 1;
            max_depth = tree[i].right -> depth + 2;
        }
        i++;
    }
    scanf("%d", &i);
    while(i--) {
        scanf("%d", &l);
        r = l;
        while (l <= max_depth) {
            for(k=0; k<num; ++k)
            {
                if( tree[k].depth == l)
                {
                    temp = tree[k].left;
                    tree[k].left = tree[k].right;
                    tree[k].right = temp;
                }
            }
            l = l + r;
        }
        inorder(tree);
        printf("\n");
    }
    return 0;
}
```

Sol.1

Output:-

Input:-

```
3
2 3
-1 -1
-1 -1
2
1
1
```

Output:-

```
3 1 2
2 1 3
```