COP3502 3/25/2022

(1) Papers hand back

(2) Binary Trees today => AVL, Tries (3)

Q1 - Tracing (last bin tree nodes)

Analyze code structure to get the picture of what it's doing

OR

Go line by line literally + trace.

50 → 2 mod 3 DO L, R

Done
- 20 → 2 mod 3 DO L, R
  - 10 → 1 mod 3 Print 12, R
    - 14 → 2 mod 3 DO LR
      - 16 → Prints 18
- 30 → Print 30, Left (Nothing)

PRINT = 12, 18, 30, 72, 90, 87

70 → 1 mod 3, Print 72, Right

90 → 0 mod 3, Print 90, Left

85 → 1 mod 3, Print 87, Right

Q2 - Skip

Q3 - Skip

Q4 - Skip

```
struct node* largest (struct node* B) {
    if (B == NULL) return NULL;
    if (B -> right == NULL)
        return B;
    return largest (B -> right);
}
```

Run-time — 1 branch of BST

$O(h)$, h = height

Worst case height = n, n = nodes

Avg case case height = $O(\lg n)$

This function best case $O(1)$.

Code both L, R $\Rightarrow$ $O(n)$, visits each node once.

Q?: One child

```
int one (struct tree_node* p) {
    if (p == NULL) return 0;
    int numtrees = 0;
    numtrees += (p -> left != NULL);
    numtrees += (p -> right != NULL);
    int res = (numtrees == 1);
    return res + one(p -> left) + one(p -> right);
}
```

Recommend : k^{th} SMALLEST *