# COP 3502 10/22/25

1) Quick Sort Avg Case Analysis
2) Binary Trees

---

Assumption: each element is equally likely to be chosen as a partition element during partition.

QS(n) low, high

1. Partition n steps — n +
   x = partition
2. QS(low, x-1)
3. QS(x+1, high)

} sizes very!

Let T(n) = qs run time avg case

| left | right | prob |
|---|---|---|
| 0 | n-1 | $\frac{1}{n}$ |
| 1 | n-2 | $\frac{1}{n}$ |
| 2 | n-3 | $\frac{1}{n}$ |
| ⋮ | | ⋮ |
| n-1 | 0 | $\frac{1}{n}$ |

$$T(n) = \underbrace{n}_{\text{Partition}} + \frac{1}{n}\left(T(0)+T(n-1)\right) + \frac{1}{n}\left(T(1)+T(n-2)\right) + \cdots + \frac{1}{n}\left(T(n-1)+T(0)\right)$$

$$T(n) = n + \frac{1}{n}\sum_{i=0}^{n-1}\left(T(i) + T(n-1-i)\right)$$

$$T(n) = n + \frac{1}{n}\sum_{i=0}^{n-1} T(i) + \frac{1}{n}\sum_{i=0}^{n-1} T(n-1-i)$$

$T(0) = 0$
$T(1) = 1$
init $a_n$

$$T(n) = n + \frac{1}{n}\sum_{i=0}^{n-1} T(i) + \frac{1}{n}\sum_{i=0}^{n-1} T(i)$$

$$T(n) = n + \frac{2}{n}\sum_{i=0}^{n-1} T(i)$$

Let $S(n-1) = \sum_{i=0}^{n-1} T(i)$

$$T(n) = n + \frac{2}{n} S(n-1) \quad \checkmark$$

$$nT(n) = n^2 + 2S(n-1) \quad \checkmark$$

$$-\ (n-1)T(n-1) = (n-1)^2 + 2S(n-2) \quad \checkmark$$

$S(n-2) = \sum_{i=0}^{n-2} T(i)$

---

$$nT(n) - (n-1)T(n-1) = (2n-1) + 2\left(\sum_{i=0}^{n-1} T(i) - \sum_{i=0}^{n-2} T(i)\right)$$

$$nT(n) - (n-1)T(n-1) = (2n-1) + 2T(n-1)$$

$$\frac{nT(n)}{n(n+1)} = \frac{(n+1)T(n-1)}{n(n+1)} + \frac{(2n-1)}{n(n+1)}$$

$$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{(2n-1)}{n(n+1)}$$

Let $V(n) = \frac{T(n)}{n+1}$

$$V(n) = V(n-1) + \frac{(2n-1)}{n(n+1)}$$

$$V(n) \le V(n-1) + \frac{2n}{n(n+1)}$$

$$V(n) \le V(n-1) + \frac{2}{n+1}$$

$$X(n) = X(n-1) + f(n)$$

$$X(n) = \sum_{i=1}^{n} f(i) + f(0)$$

$$\leq V(n-2) + \frac{2}{n} + \frac{2}{n+1}$$

$$\leq V(n-3) + \frac{2}{n-1} + \frac{2}{n} + \frac{2}{n+1}$$

$$\leq V(0) + \frac{2}{2} + \frac{2}{3} + \cdots \frac{2}{n+1}$$

$$V(n) \leq \sum_{i=1}^{n+1} \frac{2}{i} = 2 \sum_{i=1}^{n+1} \frac{1}{i} \quad H_{n+1}$$

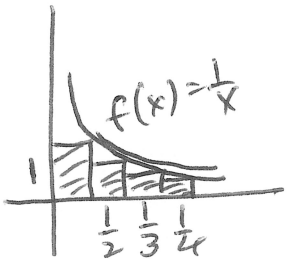
$n+1$ st Hermonic #

$H_n \sim \ln n$

$$V(n) = O(\lg n)$$

$$T(n) = (n+1) V(n)$$

$$= O(n \lg n)$$




$$H_n \leq 1 + \int_1^n \frac{1}{x} dx$$

$$\ln x \Big|_1^n$$

$$\ln n - \ln 1$$

# Binary Trees

Issues w/ dyn sized arrays + LL

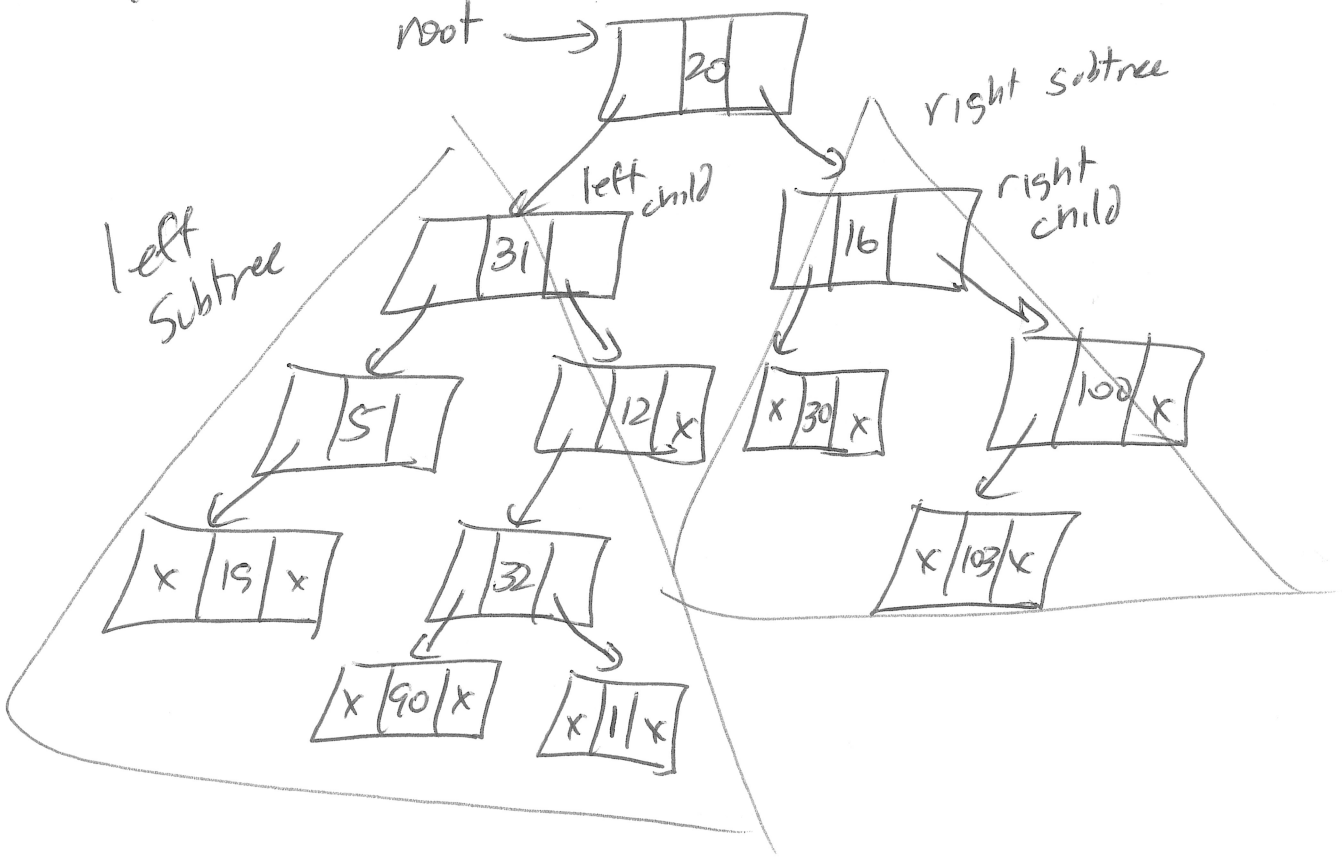
O(n) time to find stuff, adding can be O(n)

deleting can be as well.

```
struct bintreenode {
    int data; // any ptr to struct
    struct bintreenode * left;
    struct bintreenode * right;
};
```

How diff than doubly linked list?

HOW THE NODES ARE LINKED!!!



# How do I traverse a binary tree?

**preorder**

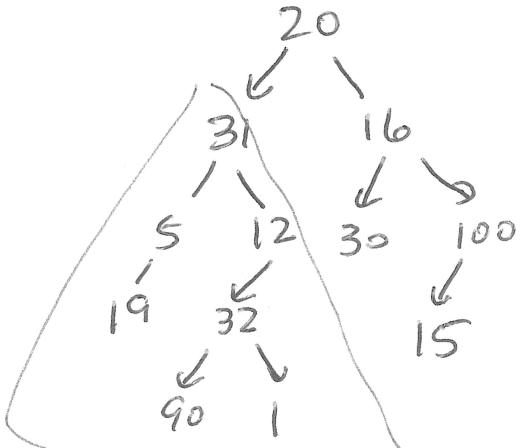
1. Print root
2. preorder(root->left)
3. preorder(root->right)

**inorder**

1. inorder(left) (root->)
2. Print root
3. inorder(root->right)

**post order**

1. postorder(root->left)
2. postorder(root->right)
3. Print root




Pre: 20, 31, 5, 19, 12, 32, 90, 1, 16, 30, 100, 15

In: 19, 5, 31, 90, 32, 1, 12, 20, 30, 16, 15, 100

Post: 19, 5, 90, 1, 32, 12, 31, 30, 15, 100, 16, 20

Q: Given Pre-order + In-order, produce the Post-order traversal.

```
void preorder(struct bintreenode* root) {

    if (root == NULL) return;
    printf("%d ", root->data);
    preorder(root->left);
    preorder(root->right);
}
// Assume create node function exists.

struct bintreenode * insert(struct bintreenode* root,
                            int num) {

    if (root == NULL)
        return makeNode(num);
    //ISSUE is DO I GO LEFT OR RIGHT?
```