Lecture 12: Divide and Conquer

- merge sort
- quick sort
- quick select
- binary search

```
Hanoi (n, src, dst, tmp):
  if n > 0:
     Hanoi (n-1, src, tmp, dst)
     move (src, dst)
     Hanoi (n-1, tmp, dst, src)
```

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

$$|\epsilon| = 0$$
$$|ax| = 1 + |x|$$

```
def len(s):
   if s == "":
      return 0
   a, x = split(s)
   return 1 + len(x)
```

$s = ax \quad a \in \Sigma$

$$T(n) = T(n-1) + c = (n+1)c = \Theta(n)$$
$$T(0) = c$$
$$T(1) = T(0) + c = 2c$$
$$T(2) = T(1) + c = 3c$$
$$T(3) = T(2) + c = 4c$$

$T(n) = (n+1)c = \Theta(n)$

$T(n) \in O(n)$ — $\exists k, a$ s.t. $\forall n > k$
$T(n) \leq a \cdot n$

$n \in O(T(n))$ $\exists k', b$ s.t. $\forall n > k'$
$n \leq b \cdot T(n)$

$\exists a, b, k$ s.t. for $n > k$

$a \cdot n \leq T(n) \leq b \cdot n$

Let n = # of hours studying
Let g = grade in 374

$g \in O(n)$

$\exists k, a$ s.t. if $n > k$
$g \leq a \cdot n$

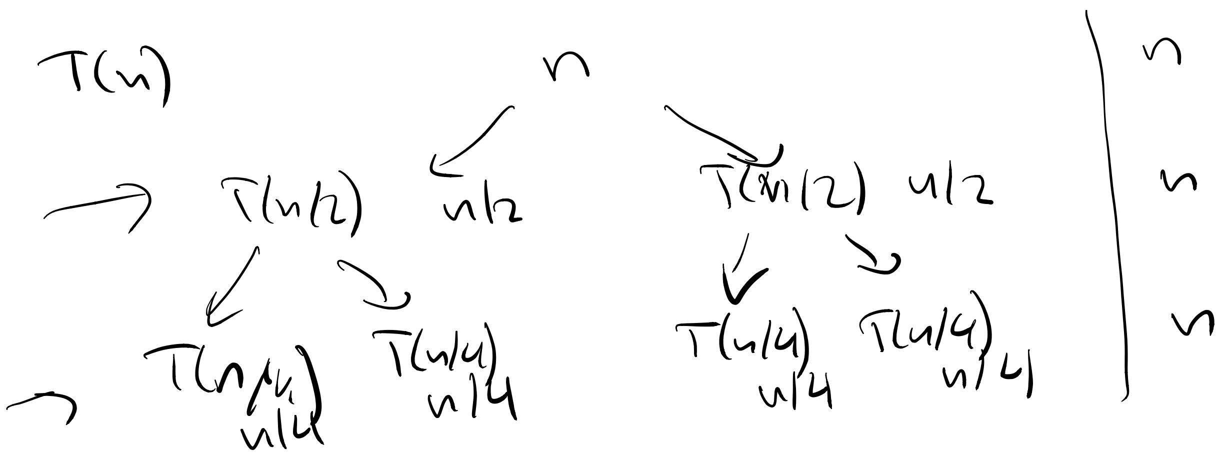
$k = 10^9$
$a = 100$

$Hanoi(n) = \Theta(2^n)$

Merge Sort

- Sort first half of array $\quad T(n/2)$
- sort second half $\quad T(n/2)$
- "sorted merge" $\quad \Theta(n)$

$T(n) = 2T(n/2) + \underline{\Theta(n)} + c$



$\log_2 n$

$$\sum_{i=1}^{\log_2 n} n = n \cdot \log_2 n = \Theta(n \log n)$$

$$= \underbrace{n + n + n \ldots + n}_{\log_2 n \text{ times}}$$

$$T(n) = T(n-1) + \Theta(n) \qquad = \Theta(n^2)$$

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

QS (Hoare) $\qquad T(n)$

pivot = x[0]

split array into first_half (< pivot) | $\Theta(n)$

second_half ($\geq$ pivot)

sort (first_half) $\quad T(n-1)$

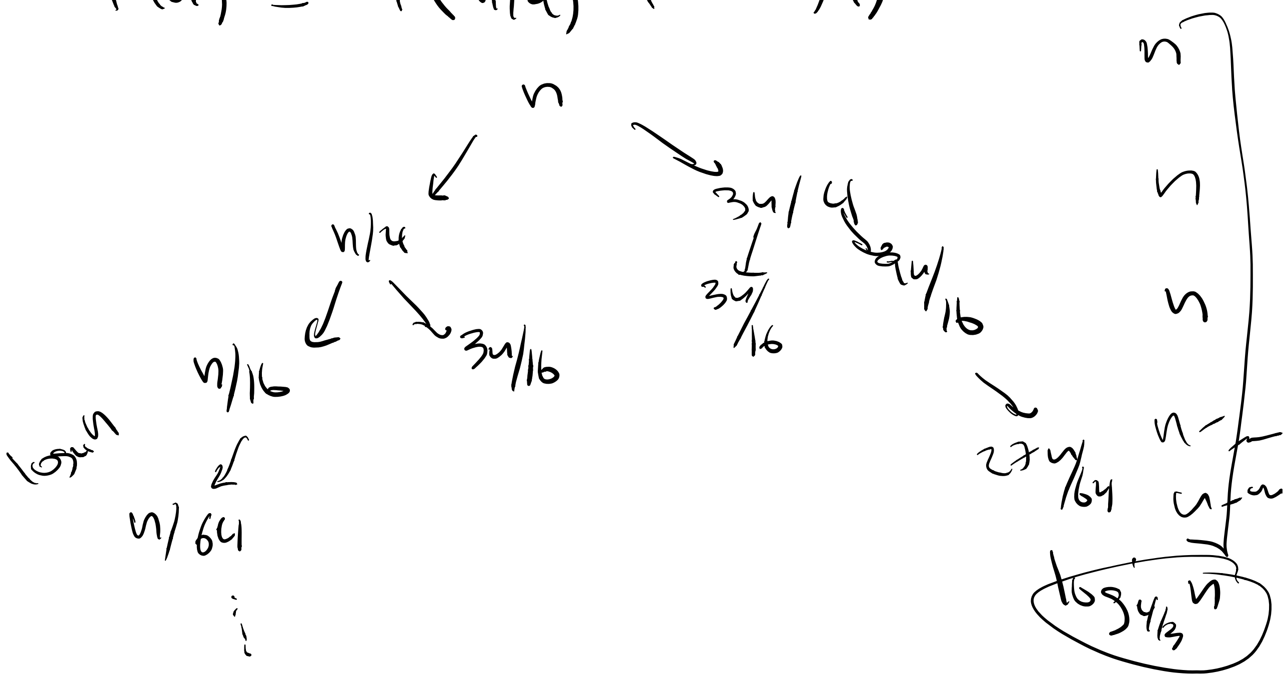
(second_half) $\quad T(n-1)$

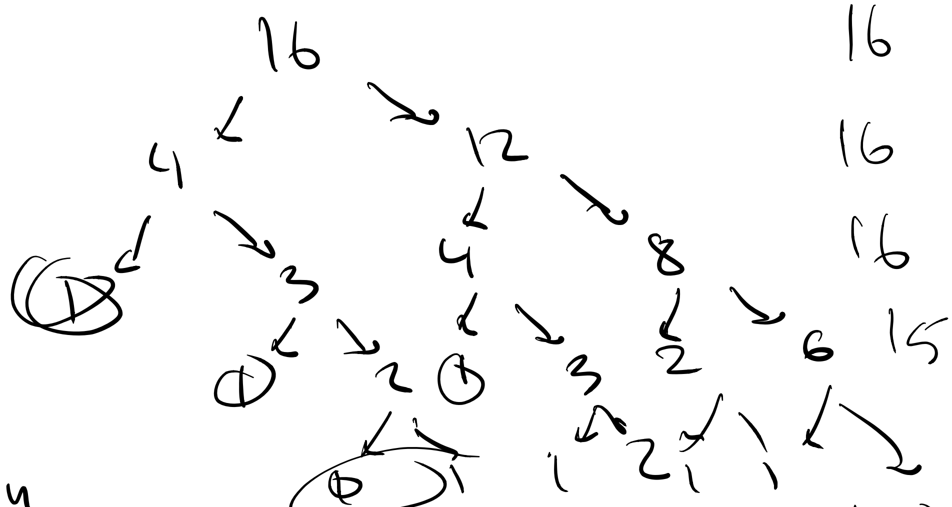
return first_half + pivot + second_half

$$T(n) = T(n-1) + \Theta(n) \qquad \Theta(n^2)$$

~~pivot~~ $n/4 \leq \text{size}(\text{first\_half}) \leq 3n/4$

$$T(n) = T(n/4) + T(3n/4) + n$$





$$\leq \sum_{i=1}^{\log_{4/3} n} n = \log_{4/3} n = \Theta(n \log n)$$