RANDOMIZED ALGO

- SELECTION
- QUICKSORT

selection: Given a set S of integers, in unsorted order, and int k, return the k^{th} smallest # in S.

GO TO SOLUTION: ① SORT SET S $n \log n$

② check position $S[k]$ $O(1)$

N AFTER THIS CLASS: Deteriministic Selection $O(n)$

RANDOMIZE ALGO: ① $v = RAND() \% |S|$

② create S_L, S_v, S_R

③ Recurse on the appropriate set

$$\text{return} \begin{cases} \text{selection}(S_L, k) & \text{if } |S_L| \geq k \\ v & |S_L| < k \leq |S_L| + S_v \\ \text{selection}(S_R,\ k - |S_L| - |S_v|) \end{cases}$$

$$T(n) = \underbrace{O(1)}_{\text{selecting pivot}} + O(n) + T(|S_L| \text{ or } |S_v|)$$

$S = \{ \underline{20}, 19, 18, 17, 16, 15, 14, 13, 12, 11 \}$ $k = 6$th. $|S| = n$

$v = 20$

$S_L = \{19, 18, 17, 16, 15, 14, 13, 12\}$ selection $(S_L, 6)$ $|S_L| = n-1$

$S_v = 20$

$S_R = \emptyset$

$S = \{ \underline{19}, 18, 17, 16, 15, 14, 13, 12, 11 \}$ $k = 6$ selection $(S_L, 6)$ $|S_L| = n-2$

$v = 19$

$S_L = 18, 17, 16, 15, 14, 13, 12, 11$ ~~selecta~~

$S_v = 19$

$S_R = \emptyset$

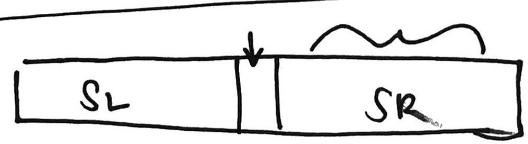
$T(n) = T(n-1) + O(n) \Rightarrow O(n^2)$

$n \rightarrow O(n)$

$\sum_{n=0}^{n-1} O(n) = O(n^2)$

$O(n-1)$ $\left[\begin{array}{c} n \\ | \\ n-1 \\ | \\ n-2 \end{array} \right.$

IDEAL



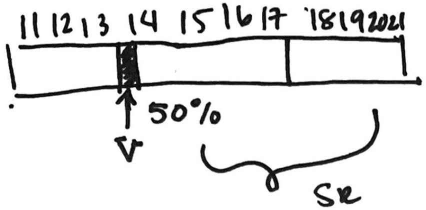
n #'s to get the middle

$\frac{1}{n}$

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

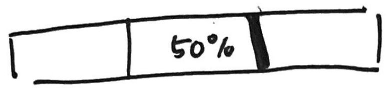
$O(n^{\log_2 1})$ vs $O(n)$

$O(n^0)$ vs $O(n^1)$ $\Rightarrow O(n)$



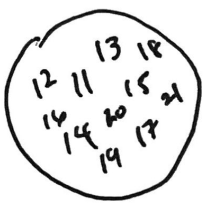
$|S_R| = \cancel{\tfrac{n}{4}} + \frac{n}{2} + \frac{n}{4} = \frac{3}{4}n$

$|S_L| = \frac{1}{4}n$ $\Rightarrow$ SR worst $\Rightarrow$ Recurce on $T\left(\frac{3n}{4}\right)$



$S_L = \frac{3}{4}n$ $\Rightarrow$ SL worst

$S_R = 1/4n$



$\frac{1}{2}$

Calc. expected # of times I have to select a ~~#~~ number until I see one in the middle 50%?

$$E(X) = 1 \cdot \frac{1}{2} + (1 + E(X))\,\frac{1}{2}$$

↑ # of times until

↑ # not in the 50%

$$E(X) = .5 + (1 + E(X)).5$$

$$E(X) = .5 + .5 + .5E(X)$$

$$\frac{\cancel{.5}E(X)}{\cancel{.5}} = \frac{1}{.5}$$

$$E(X) = 2$$

~~RAND % (SEL) # ORACLE checks if # is in 50%~~ 0 BAD #

$$T(n) = \underbrace{n}_{} + \underbrace{n-1}_{} + T\left(\frac{3}{4}n\right)$$

sets S_L, S_R creation, choose a GOOD #. on a bad #.

Def "good #": Is # that is in middle 50% if S is sorted

$T(n) = \underbrace{(a-1)O(n)}_{\text{set creation for bad \#s}} + O(n) + T(\frac{3}{4}n)$

set creation for bad #s, where a is the expected # of times you choose until a "good" #

1 | n-1 | 3/4n

n | n-1 | $\frac{3}{4}n$ | $\frac{3}{4}n-1$ | $\frac{3^2}{4^2}n$

n

n-1 BAD CHOICE FOR V

$\frac{3}{4}n$

$\frac{3}{4}n-1$

$\sum_{i=0}^{\log_{4/3} n} \left(\frac{3}{4}\right)^i n \Rightarrow O(n)$

Randomize ⇒ linear

Selection (S, k) ::

V = S[RAND % |S|+1]

create S_L, S_V, S_R

Recurse S_L OR S_R OR RETURN V

Def of good is NOW MIDDLE 60%

20%	60%	20%

good

$S_R = .60n + .20n$

$= \frac{8}{10}n = \frac{4}{5}n$

$T(n) = T(\frac{4}{5}n) + O(n) + O(n)$

⇒ BY MASTERS THIS IS STILL $O(n)$

Analysis: $E(x) = .6(1) + .4(1 + E(x))$

↑ # times you generate pivot until good choice

$E(x) = .6 + .4 + .4E(x)$

$.6E(x) = 1 \quad E(x) = \frac{1}{.6} = \frac{10}{6} = \frac{5}{3}$

ABOUT TWO PICKS UNTILL GOOD PIVOT.

```
QUICKSORT ( A[ ], low, high) ::
    if (low<high) {
        pivot = RAND % A.size()
        RESORT ARRAY [A] AGAINST PIVOT ≡ PARTITION A
        QUICKSORT ( A[ ], low, pivot);
        QUICKSORT( A[ ], pivot+1, high);
```

smaller then pivot: $\boxed{\text{left}}$ larger then pivot: $\boxed{\text{Right}}$

level	Recursion Tree	WORK AT EACH Level
$i=0$	n	$O(n)$
$i=1$	$\boxed{1}\ \boxed{n-1}$	$O(n)$ BAD CHOICE
$i=2$	$\frac{1}{4}n \quad \left(\frac{3}{4}\right)n$	$O\left(\frac{1}{4}n\right)+O\left(\frac{3}{4}n\right)=O(n)$ GOOD CHOICE
	$\boxed{1}\ \frac{1}{4}n-1 \quad \boxed{1}\ \boxed{\frac{3}{4}n-1}$	$\Rightarrow O(n)$ BAD CHOICE
	$\left(\frac{1}{4}\right)^2 n \quad \left(\frac{1}{4}\right)\left(\frac{3}{4}n\right) \quad \left(\frac{3}{4}\right)\frac{1}{4}n \quad \boxed{\left(\frac{3}{4}\right)\left(\frac{3}{4}\right)n}$	$\Rightarrow O(n)$ GOOD CHOICE

$$\left(\frac{3}{4}\right)^{i/2} n \qquad j = i/2$$

$$\left(\frac{3}{4}\right)^{j} n = 1 \qquad j = \log_{4/3} n$$

work per level: $O(n)$

$$2\left(\sum_{j=0}^{\log_{4/3} n} O(n)\right)$$

$$2\cdot n\cdot\sum_{j=0}^{\log_{4/3} n} 1 \leq \underline{O(n\log n)}$$