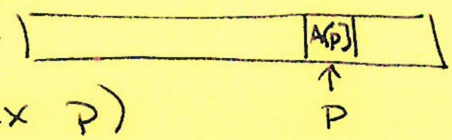
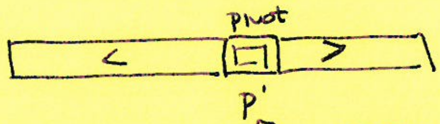
12 Feb 2026

Quicksort, at a high level:

- input: Array A = 

- If stmt check
- choose pivot (index p)

- partition A into "less than A[p]" + A[p] + "> A[p]"

A = 

may not be in the centre!

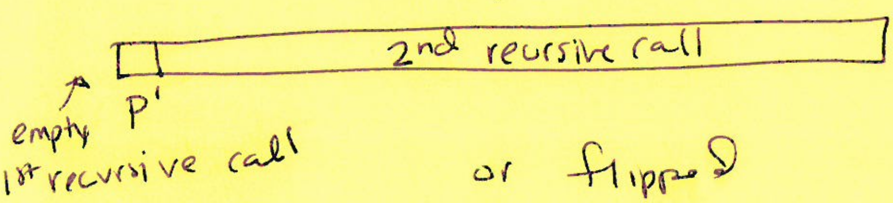
- recursive calls to Quicksort (A[1... p'-1])
  and Quicksort (A[p+1... n])

Recurrence relation: 

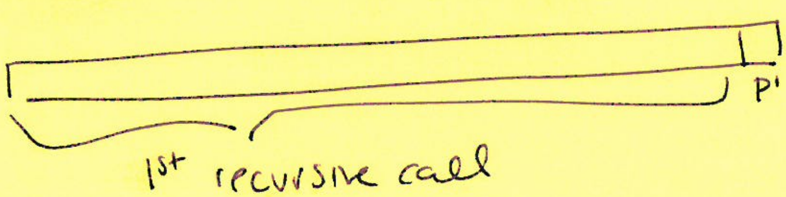
- $T: \mathbb{N} \to \mathbb{R}$ def. by $T(n) =$ # of steps to complete Quicksort (A) when $|A| = n$

- worst-case: totally unbalanced

$$T(n) = \cancel{T(0)}^{\Theta(1)} + T(n-1) + \Theta(n)$$
$$= T(n-1) + \Theta(n) \quad \in \Theta(n^2)$$



or flipped



base case called here QS(A[n... n])

Assorted Notes:

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ \hline \pi & 17 & 6.2 \end{array}$$

- Quicksort $(A[1\ldots -1])$ $\quad A([1\ldots -1]) = A([\,])$

AKA Quicksort $(A[\,])$

goes into Quicksort algo &
gets "caught" by the if stmt / base case

effect: no change to A is made

- Quicksort $(A[i])$ for $i \in \{1, 2, \ldots, n\}$

also gets caught by that if stmt

effect: no change to A is made

} 2 base cases "if$(n>1)$" executes $\forall$ except these two cases.

- Flip a coin:

Heads: you win \$2

Tails: you lose \$5

Expected value of 1 coin flip?

"collecting data"

① List my outcomes: $\{H, T\} =: \Theta$

② understand prob of each outcome

$$\mathbb{P}(H) = 1/2$$

$$\mathbb{P}(T) = 1/2$$

③ understand the value of each outcome

$$\mathbb{V}(H) = +\$2$$

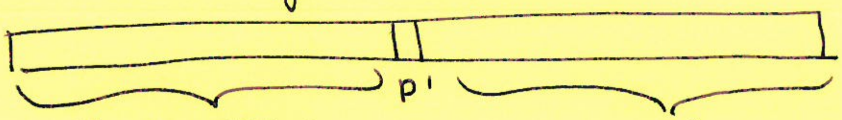
$$\mathbb{V}(T) = -\$5$$

④ $$\mathbb{E}(\overset{1}{\text{FLIP}}) = \sum_{x \in \Theta} \mathbb{P}(x) \cdot \mathbb{V}(x) = \overbrace{\mathbb{P}(H) \cdot \mathbb{V}(H)}^{H} + \overbrace{\mathbb{P}(T)\,\mathbb{V}(T)}^{T}$$

$$= 1/2 \cdot \$2 + 1/2 \cdot -\$5$$

$$= \$1 - \$2.50 = -\$1.50$$

• best-case: always balanced!



1st recursive call on $\lfloor n/2 \rfloor$ elements

2nd recursive call on $\lfloor n/2 \rfloor$ elements

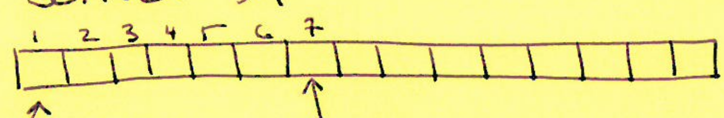
$$T(n) = T(n/2) + T(n/2) + \Theta(n)$$
$$= 2T(n/2) + \Theta(n)$$
$$\in \Theta(n \log n)$$

Ⓠ Randomized Quicksort:

the pivot is chosen by randomly selecting a pivot element (using a good random # generator)

Then, <u>all</u> elements are equally likely to be selected as the pivot.

Consider A_{OUT} = sorted A

=


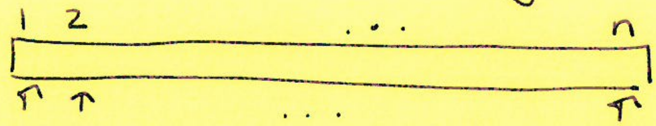
also has prob $1/n$ to be selected

p = 7 has prob $1/n$ to be selected as the pivot

$\mathbb{E}$xpected runtime for Randomized QS.

random variable i = location of pivot in the ~~sorted array~~. sorted array

① outcomes?



Any location $i \in \{1, 2, \ldots, n\}$

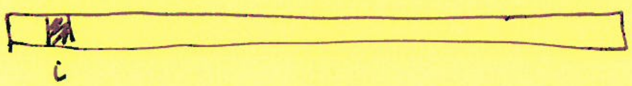
② $\mathbb{P}(i=1) = 1/n$
$\mathbb{P}(i=2) = 1/n$
$\vdots$
$\mathbb{P}(i=n) = 1/n$

} always $1/n$ b/c we have a good RNG

③ "value" is runtime

$i=1 \quad T_1(n) = T(0) + T(n-1) + \Theta(n)$

$i=2 \quad T_2(n) = T(1) + T(n-2) + \Theta(n)$



i-th case: $T_i(n) = T(i-1) + T(n-i) + \Theta(n)$

④ in expectation....

$$\mathbb{E}(T(n)) = \sum_{i=1}^{n} \mathbb{P}\binom{\text{pivot}}{\text{at } i} \cdot \mathbb{E}(T_i(n))$$

Ok... can we simplify this (asymptotically)?
Everything is proportional to the # of comparisons.

$X(n)$ = # expected of comparisons when $|A| = n$

$$\Theta(X(n)) = \Theta(T(n))$$

So, let's count comparisons

(sorted)

<u>Question</u>: ~~Are~~ What is the probability that index i & index j are compared? to each other

In current iter:

- pivot $= i$ — certainly compared!
- pivot $= j$ — certainly compared!
- pivot in between - certainly not compared!
- pivot $< i$
- pivot $> j$

} eh? They're in the same sub-problem could be, could not be. Don't know until we pick a pivot between i & j (inclusive)

$$\mathbb{P}(i, j \text{ compared}) = \frac{2}{j-i+1}$$

} all in that interval are equally likely to be that one pivot that gives us a certain answer if they are compared or not

$\mathbb{P}(i, j \text{ compared}) \overset{!!}{=} X_{ij}$

$$X_{ij} = \begin{cases} 1, & \text{if compared} \\ 0, & \text{otw} \end{cases}$$

$$X = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{P}(X_{ij}=1) \cdot \mathbb{V}(X_{ij}=1) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{2}{j-i+1}\right)(1)$$

↑ total # comparison

continued from ⑤

$$X = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{2}{j-i+1}\right)(1)$$

$$= \sum_{i=1}^{n} 2\left(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-i+1}\right)$$

$$\leq \sum_{i=1}^{n} 2\left(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}\right)$$

nth harmonic #

$$= H_n \sum_{i=1}^{n} 2$$

a const!

$$= 2n \cdot H_n \in 2n \cdot \Theta(\log n)$$

$$\in \Theta(n \log n)$$

<u>note</u>: most implementations of QS use randomized QS! ☺

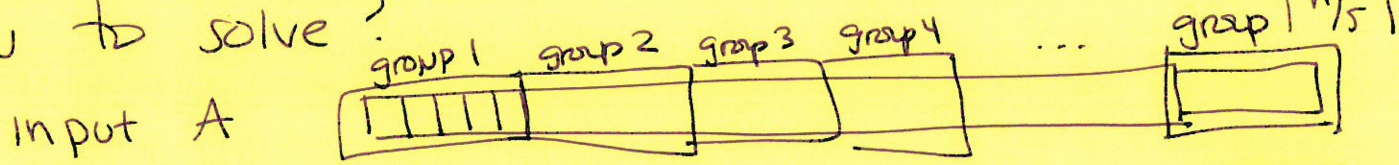
But... there is a way to pick a good value *always* in linear time.

goal: get this middle value
AKA - the median
AKA - $A(\lceil n/2 \rceil)$ = pivot
$(n/2)$nd smallest element

k-select (generalizing this)
- what is the k^{th} smallest element?

QUICKSELECT(A, $n/2$)

How to solve?



① Divide A into groups of size 5.
note: last group may be smaller
$\#\text{groups} = \lceil n/5 \rceil$ $\quad\quad \Theta(n)$

② For each group, find median m_i
time = # groups · per group comp
$= \Theta(n) \cdot \Theta(1)$
$= \Theta(n)$

③ Recursively, find the median of medians
QUICKSELECT$\left(\{m_i\}, \frac{\lceil n/5 \rceil}{2}\right)$
↑ index of median of medians

$\boxed{\text{Runtime is linear!}}$ Check out recursion tree.