Randomization

QuickSelect

Skip Lists

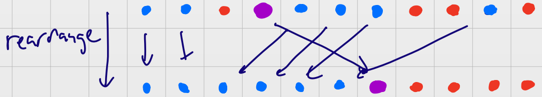
Selection

Input : array of n numbers, $x_1, \dots, x_n$
Output : the kth smallest.

Simple algorithm : sort, then return kth index $O(n \log n)$

QuickSelect is a randomized algorithm w/
expected $O(n)$ runtime
worst case $O(n)$ runtime.

Idea: Choose a random pivot.
Use divide and conquer.
Only going to recurse on one side.



If $k=4$, then only need to look in the blue numbers.

Random Variables / Expectation

Probability Space S.

A RV associates a value to each outcome.

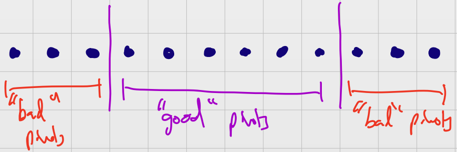
If X is a RV, it's expected value is:

$$E\{X\} = \sum_{s \in S} p(s) X_s$$

Important Property: $E[aX + bY] = aE[X] + bE[Y]$

What's the expected running time of Quick Select?

Define $T\{n,k\}$ to be the expected runtime of k-QS.
Define $T\{n\}$ to be the max over k.



The idea is that good pivots always reduce the problem size to $\le \frac{3}{4}n$, regardless of k.

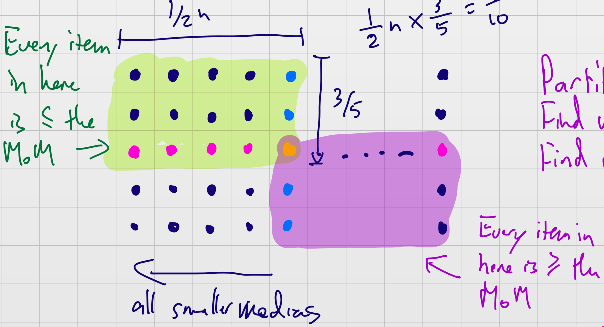
$$T\{n\} \le n + \frac{1}{2}T\{\tfrac{3}{4}n\} + \frac{1}{2}T\{n\}$$

$$\Rightarrow \frac{1}{2}T\{n\} \le n + \frac{1}{2}T\{\tfrac{3}{4}n\} \Rightarrow T\{n\} \le 2n + T\{\tfrac{3}{4}n\} \Rightarrow T\{n\} = O(n)$$

QuickSelect in worst case $O(n)$.

Median of medians

$$\frac{1}{2}n \times \frac{3}{5} = \frac{3}{10}n$$



1/2 n
3/5
Every item in here is ≤ the MoM
all smaller medians
Every item in here is ≥ the MoM

Partition into groups of 5
Find median for each group
Find median of medians
(recursively)

The MoM is ≥ 30% and ≤ 30%
⇒ good pivot!

Median of Medians

1. Partition into groups of 5	$O(n)$
2. Median of each group	$O(n)$
3. Recurse to find MoM	$T(\frac{n}{5})$
4. Recursively run QS	$T(\frac{7}{10}n)$

$$T(n) = O(n) + T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right)$$

$\Rightarrow T(n) = O(n)$ by Master Theorem

14	14
18	
23	
28	
34	34
42	42
50	
59	
66	
72	72

Sorted linked list

$14 \rightarrow 18 \rightarrow 23 \rightarrow 28 \rightarrow 34 \rightarrow 42 \rightarrow$

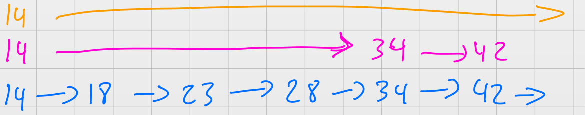
Bad data structure!

$O(n)$ insert

$O(n)$ query

14	14
18	
23	
28	
34	34
42	42
50	
59	
66	
72	72

Sorted linked list



Bad data structure!

O(n) insert

O(n) query

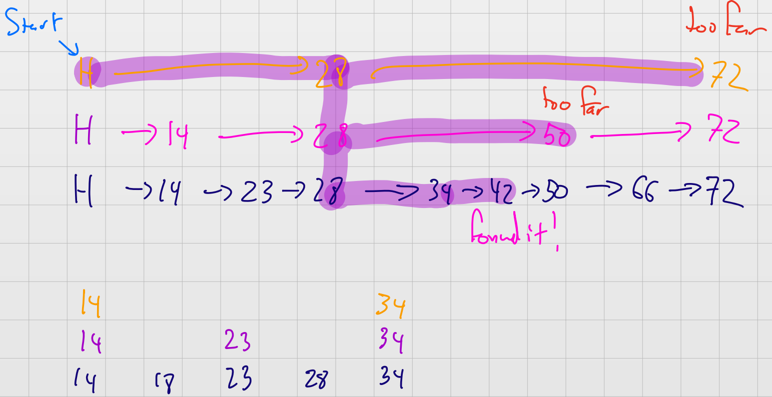
Skip List

Have 2log n levels to store n items.

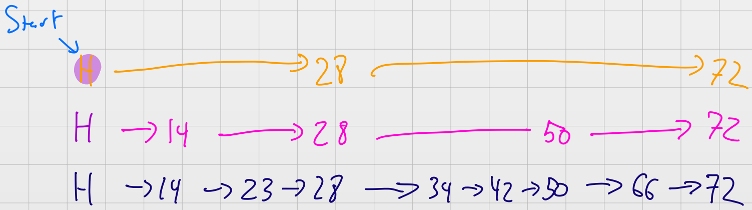
Each level is a sorted linked list

When an item is inserted, flip a coin to determine whether it gets upgraded, if so, repeat.

Expected Search Cost?



Expected Search Cost?


Start
H → 28 → 72
H → 14 → 28 → 50 → 72
H → 14 → 23 → 28 → 34 → 42 → 50 → 66 → 72