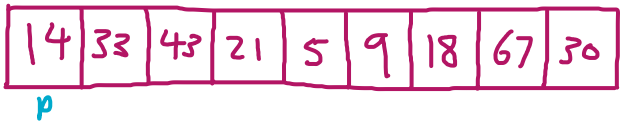
QuicKsort

(1) Choose a "pivot"

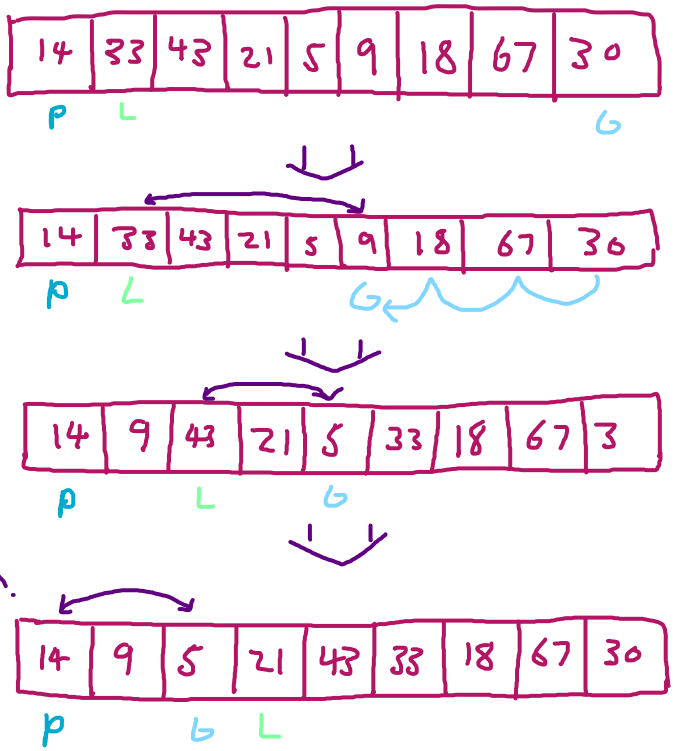
In this note, our pivot-picking strategy chooses first element as pivot.



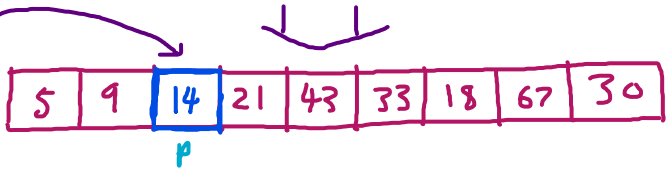
(2) Partition: run Tony Hoare's In-place Partitioning scheme

- Partition = place all items less than or equal to the pivot to the left and all items greater than pivot to the right.
- Tony Hoare: accomplishes this in $\Theta(N)$ time and $\Theta(1)$ space.

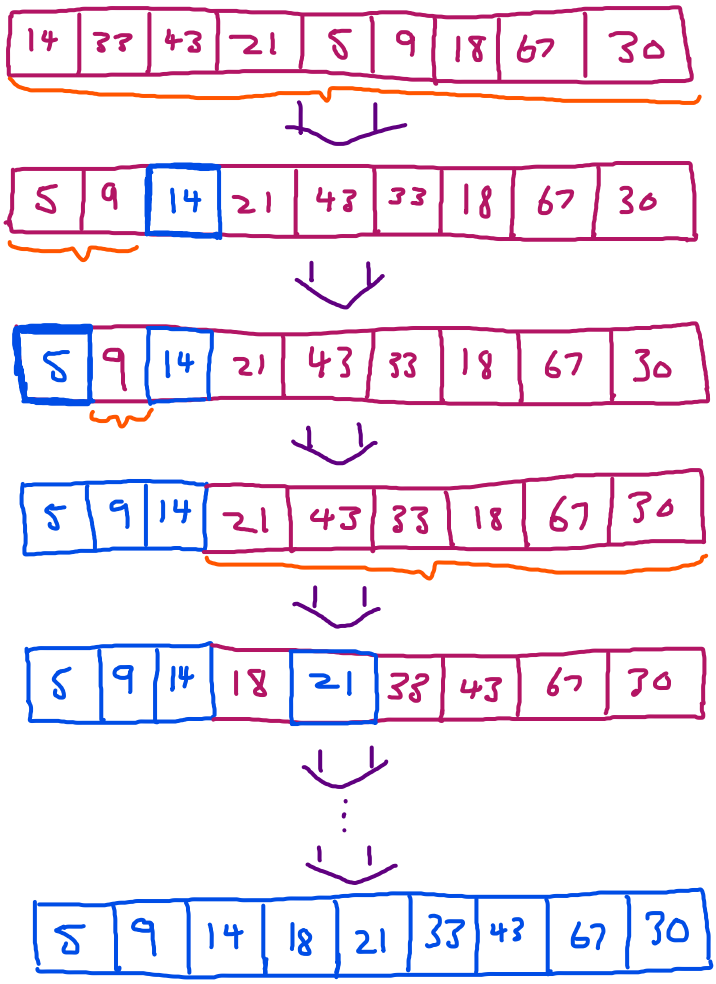
a) Place L pointer at index 1.
b) Place G pointer at end.
c) Move pointers towards each other until L lands on an item that is greater than pivot and when G lands on an item less than or equal to pivot.
   Swap items L and G are pointing to.
d) Repeat (c) until L is on right side of G.
e) Swap pivot and item at G.



Notice: the pivot is now in the correct position of the sorted array

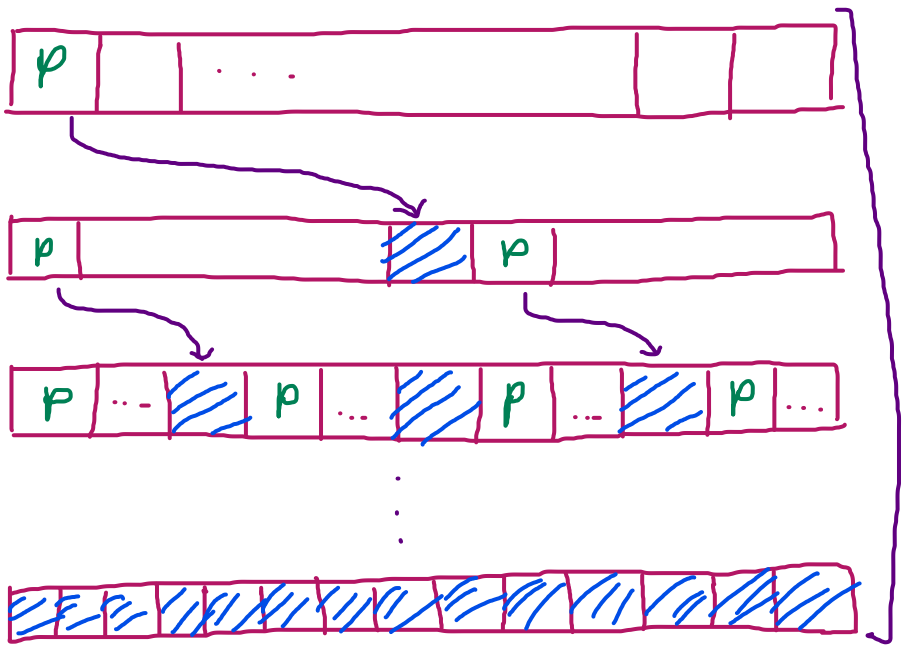


(3) Recursion: repeat steps 1 and 2 on left and right subarrays of pivot.



Runtime?

Best case: $\Theta(N \log N)$

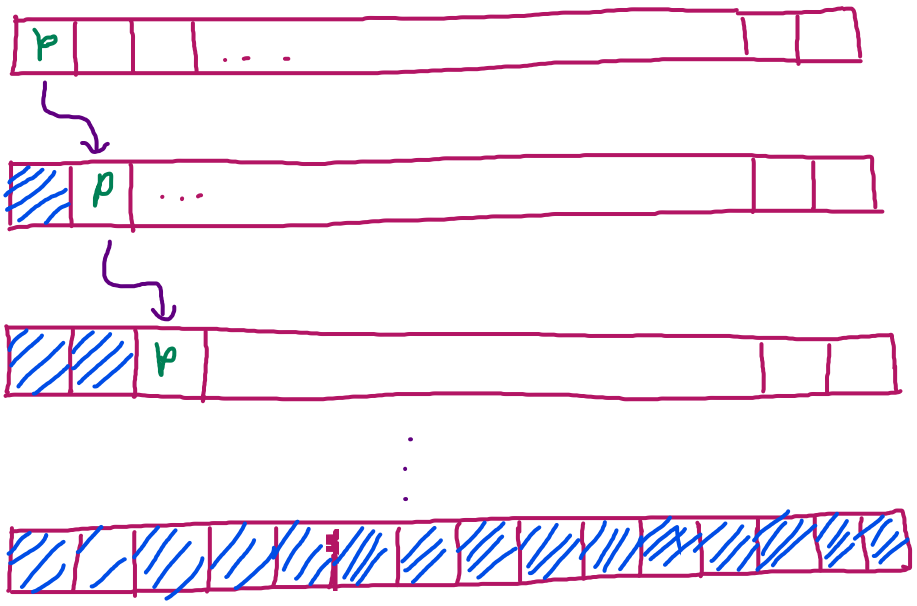


How many elements involved in partition

Level	Work
0	N
1	$\frac{N}{2} * 2 = N$
2	$\frac{N}{4} * 4 = N$
⋮	$< N$
$\log N$	$< N$

$\underbrace{N + N + \ldots + N}_{\log N} = \Theta(N \log N)$

Worst case? $\Theta(N^2)$



Level	Work
0	N
1	N-1
2	-2
⋮	⋮
N	1

$N + (N-1) + (-2) + \ldots + 2 + 1 = \Theta(N^2)$