Q⇒ Given an unsorted array arr of length n. Also given a pivot element p. Rearrange the array such that all the elements lesser than p are to the left of p & everything greater p are to the right of it.
(Arrangement of left & right elements doesn't matter).

Ex

[1, 6, 3, 8, 9, 2, 5]
  0 1 2 3 4 5 6

p ⇒ 2 index

p ⇒ 2

↳ elem ⇒ 3

→ [1, 2, 3, 6, 8, 9, 5]

arr[R] = pivot

Swap

Swap [i+1, R]

L → 0

R → n-1

[1, 2, 3, 8, 9, 6, 5]

i

pivot-index → 2

pivot-element → 3

```
if (arr[j] <= pivot_element)
        i++
        swap(arr, j, i);
}
```

i → everything to the left of i is less than pivot or equal to pivot

L=0

R=n-1

[23, 9, 18, 32, 50, 61]

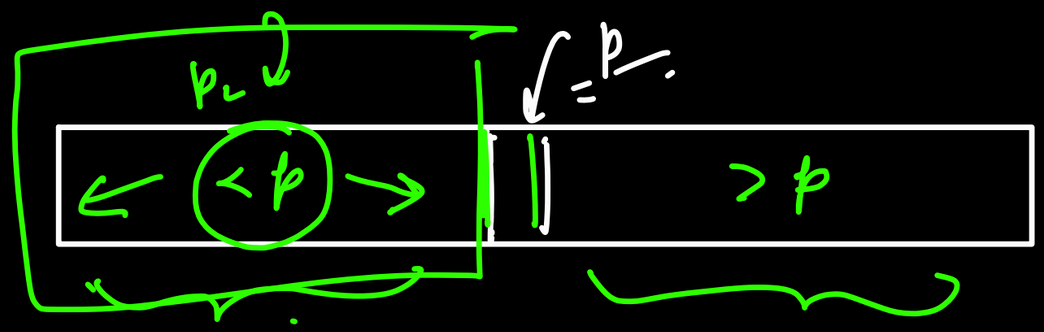
i

pivot_index = 1

pivot_elem = 32

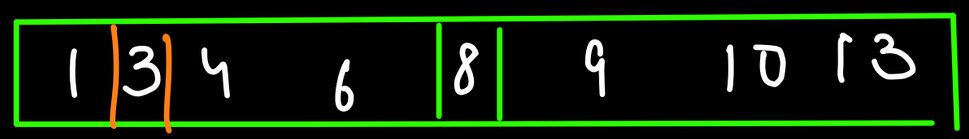
O(n) time

O(1) space



Quick Sort

2 pivot ⇒ 3



$f(arr, l, r) =$ partition$(l, r,$ pivotindex$)$

$f(arr, l,$ pivotindex $-1)$

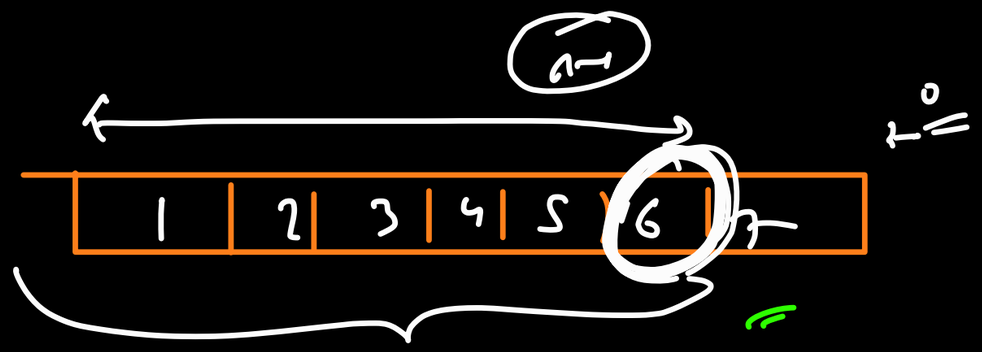
$f(arr,$ pivotindex $+1, r)$

Quicksort

the index on which pivot resides after partition

//

$O(n^2)$

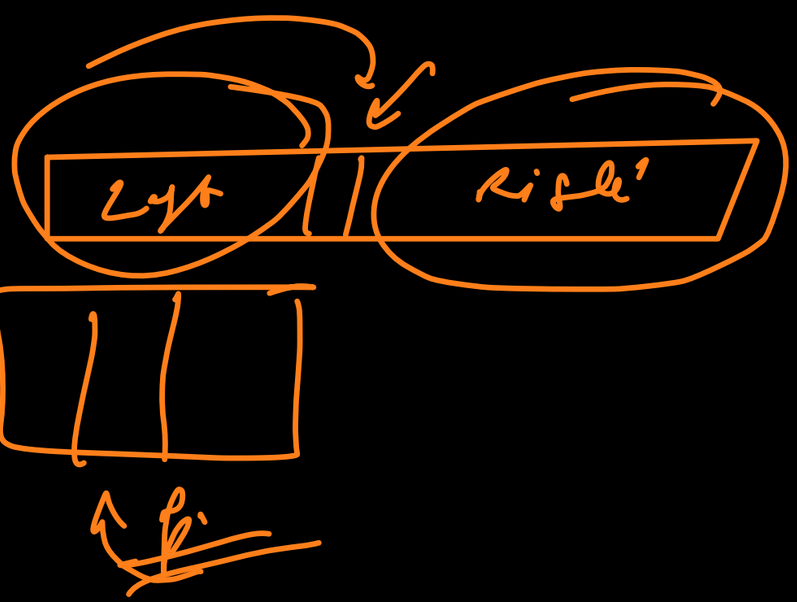


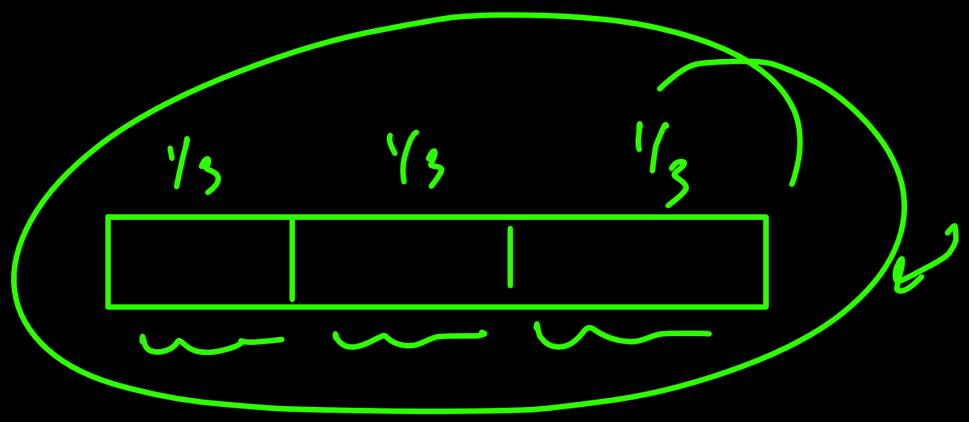
Randomized quick sort

$O(n \log n)$

$n-1 + n-2 + n-3 \ldots\ldots$

$O(n^2)$





2 last elem as pivot

$$T(n) = T(n-1) + T(0) + O(n)$$

↙

no. of ops to do quick sort on a n-1 array

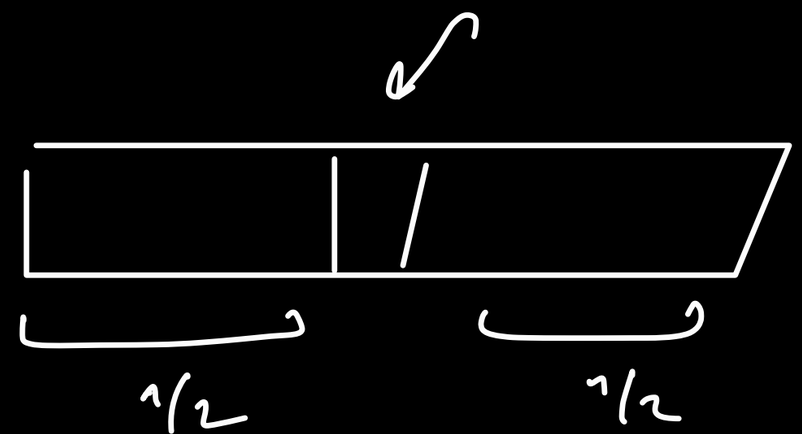
$$T(n) = T(n-1) + O(n)$$

$$= T(n-2) + O(n) + O(n)$$

$$= T(n-3) + O(n) + O(n) + O(n)$$

$$\vdots$$

$$\Rightarrow T(1) \longrightarrow O(n^2)$$



$$T(n) = T(n/2) + T(n/2) + O(n)$$

$$T(n) = 2T(n/2) + O(n)$$

Best Case

$$\Omega(n \log n)$$

	Merge	Quick
Time →	$\Omega(n \log n)$	$\Omega(n \log n)$
	$\Theta(n \log n)$	$\Theta(n \log n)$
	$O(n \log n)$	$O(n^2)$
Space →	$O(n)$	$O(\log n)$
In place →	Not	Yes
Stability	Yes	NO