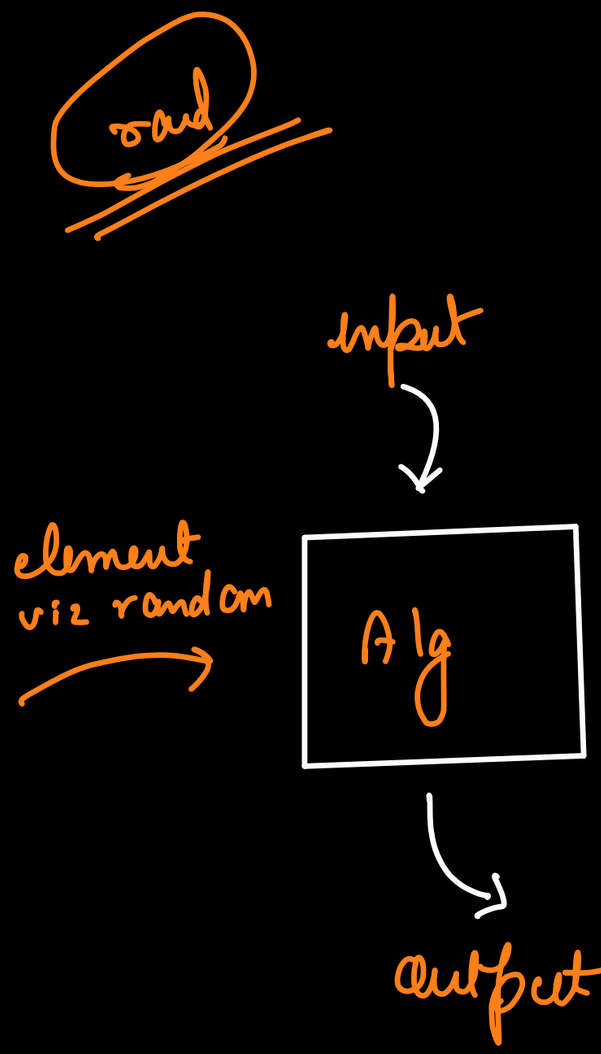
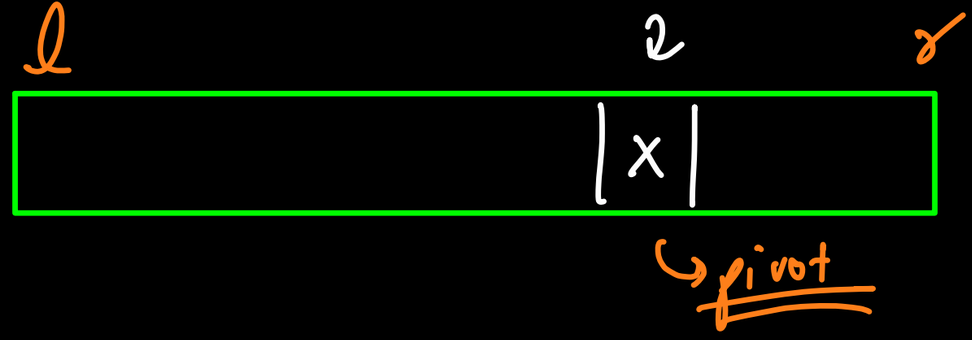
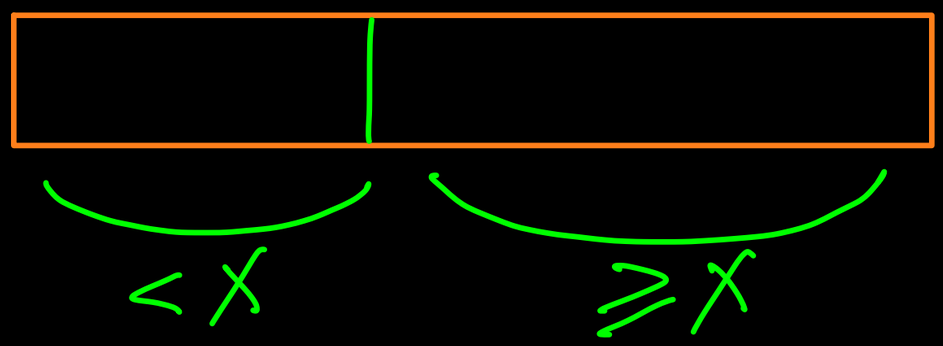


Quick Sort → Randomization

How quick sort works actually ???

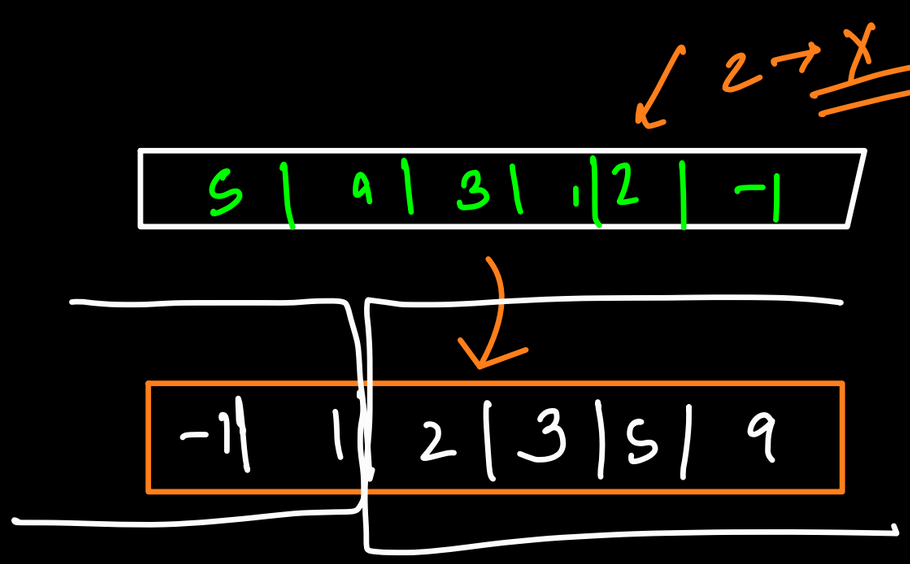


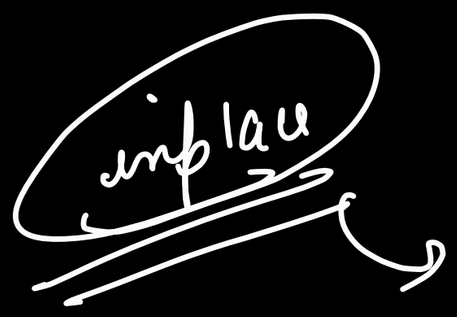
$x = arr[\ random(l, r)\ ]$

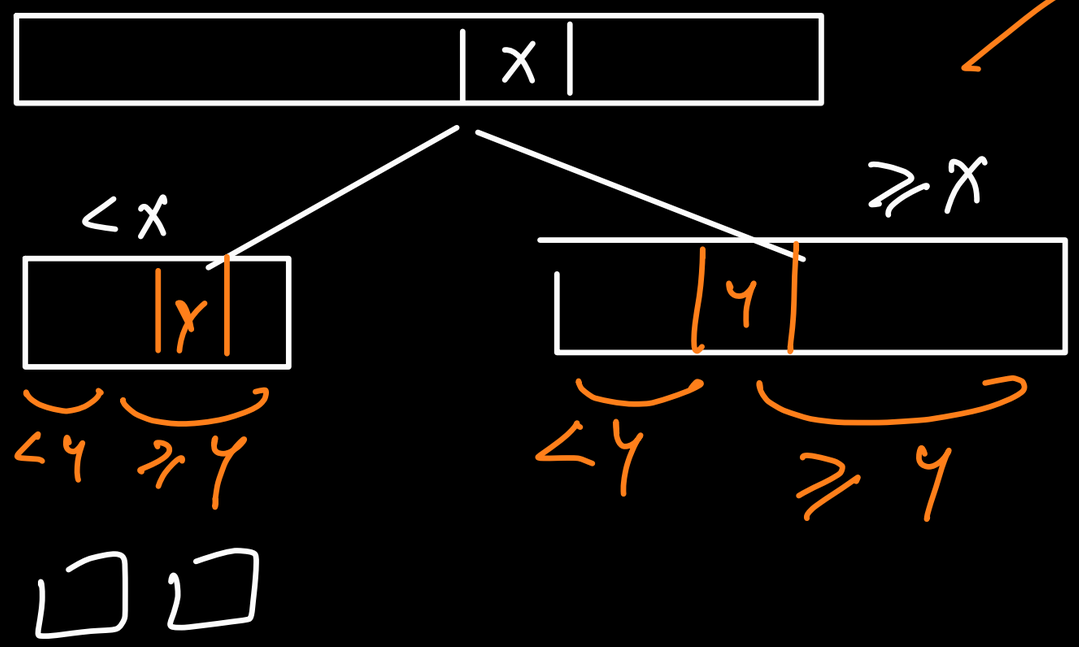


We segregate elements in the array into 2 parts such that left part has all the elements less than x & right has all element greater or equal to x.

left and right part might not be equal halves.

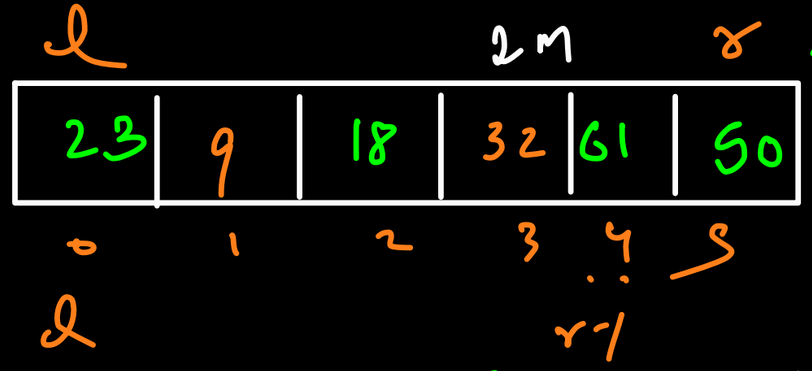






We won't be making new arrays & segregate everything in the same array.

Partition algorithm



pivot → X = 32

pivot index → m

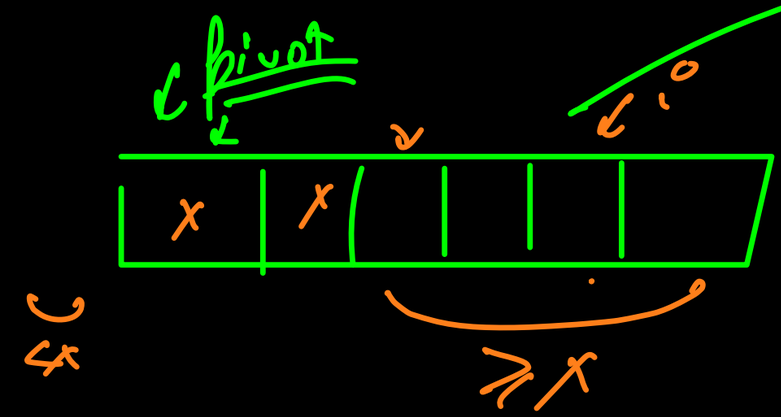
randomly

m → correct index for pivot

loop → (l, r-1)

[0, 2] → < X

[3, n-1] → ≥ X

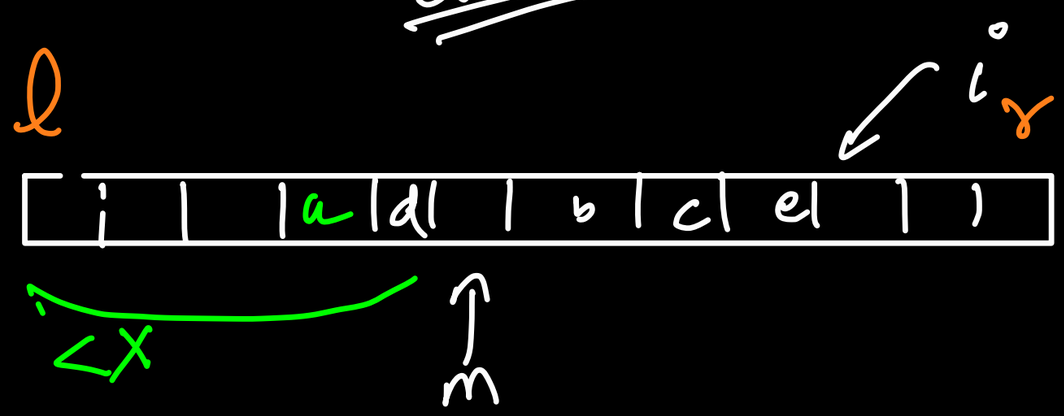


Two pointers

$d < x$

$a < x$

swap (i, m)



m++

all the elements present on the indexes $< m$ are for sure $< x$.

```
partition (arr, l, r, Xind) {
    pivot = arr[Xind]
    Swap (arr, Xind, r);
    m = l
    for (i = l; i <= r-1; i++)
        if (arr[i] < pivot) {
            Swap (arr, i, m);
            m++;
        }
    }
    Swap (arr, m, r);
}
```

→ $O(n)$

→ space $\rightarrow O(1)$

$T(n) = T(n-1) + c_1$

↳ Faulty

Quicksort on a list

$T(n) = T(n-2) + c(n-1)$

$T(n-2) = T(n-3) + c(n-2)$

$T(n-3) = T(n-4) + c(n-3)$

$\vdots \quad \vdots \quad \vdots$

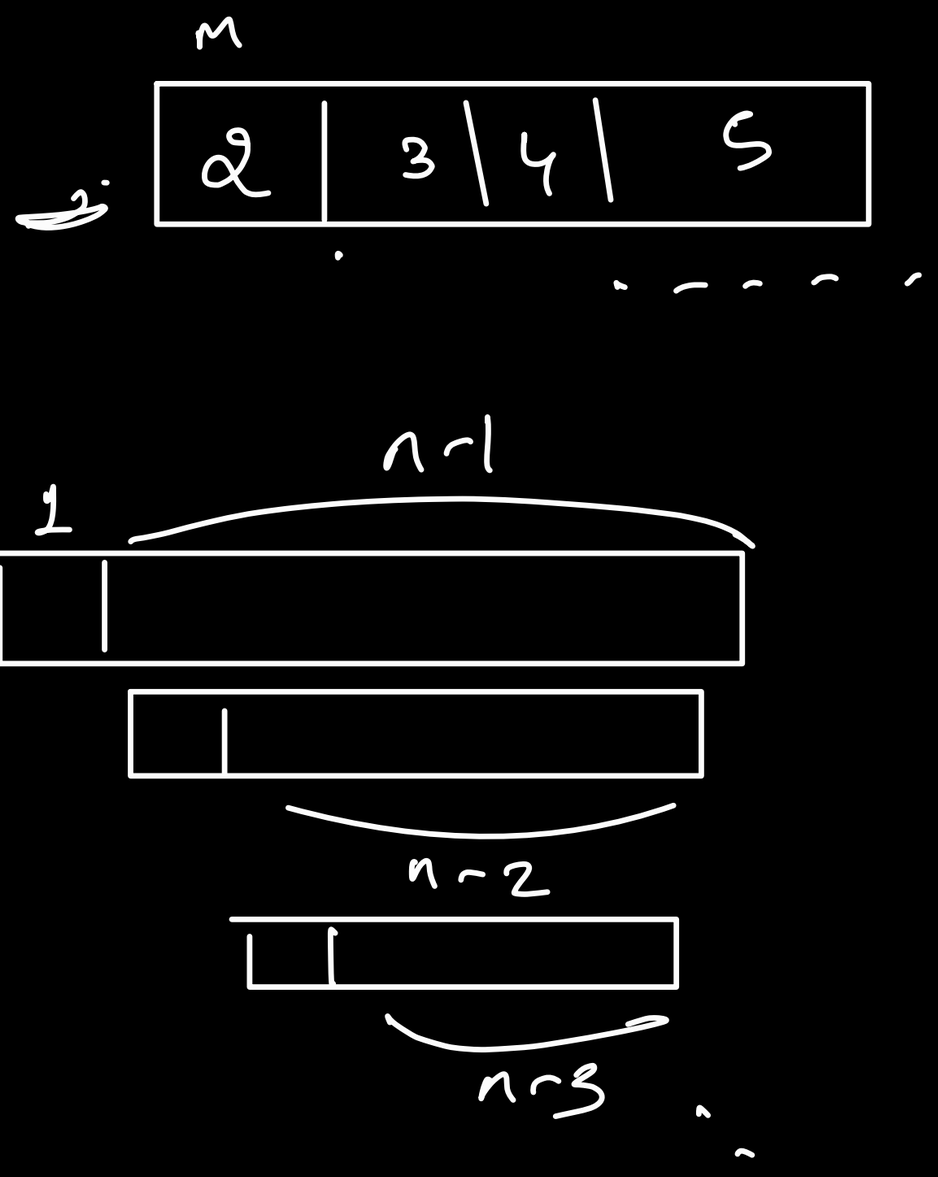
$T(2) = T(1) + c \times 2$

$T(1) = T(0) + c$

$T(n) = T(0) + c(n-1) + c(n-2) + c(n-3) \cdots \cdots c$

$T(n) = c + c\left(\frac{n \times (n-1)}{2}\right) \simeq O(n^2)$

n+1, 6



n

n-1

n-2

Contradiction

$O(n^2)$

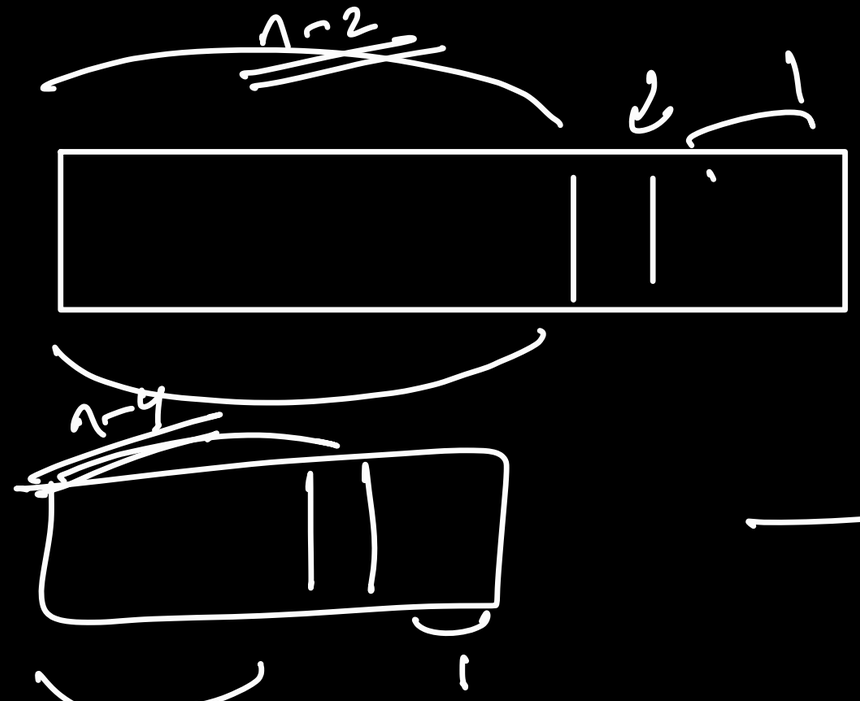
Worst Case

OR pivot → largest

2 pivot

→ n-1
n-2
n-3

$O(n^2)$



n

n-2

n-4

n-6

$O(n^2)$

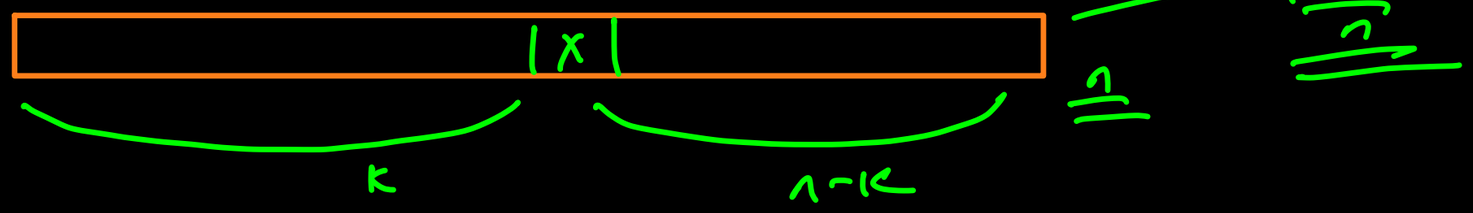
Time Complexity Analysis for QuickSort

↳ Randomised

$T(n) \rightarrow$ # of ops

↓

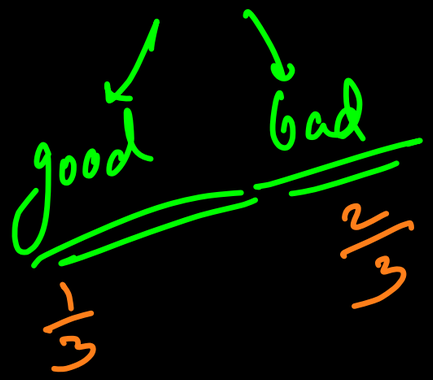
# of operations

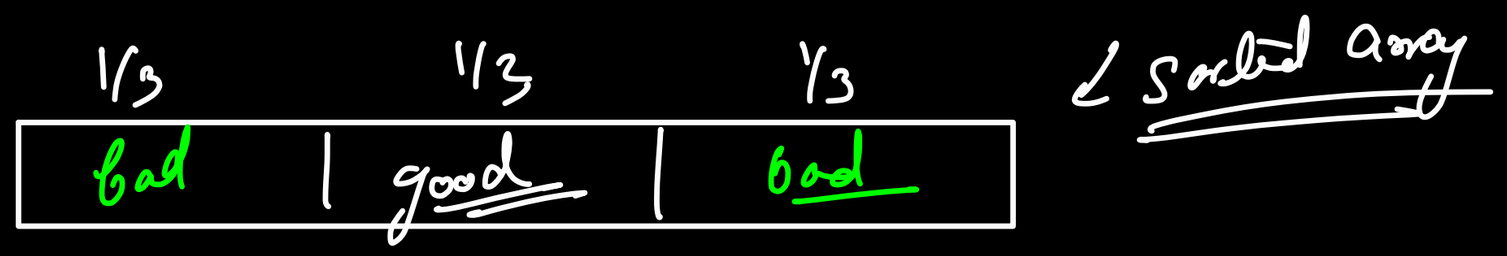


$$E(T(n)) = \frac{1}{n} \times \sum_{k=0}^{n-1} \left(n + T(k) + T(n-k) \right)$$

↳ $T_k(n)$

$$E(T(n)) = \frac{1}{n} \times T_0(n) + \frac{1}{n} \times T_1(n) + \frac{1}{n} \times T_2(n) \ldots\ldots$$





↳ sorted array

↳ what is the best pivot for quicksort ??

↳ median

$$\frac{1}{n} \times \sum_{k=0}^{n-1} \left(n + T(k) + T(n-k) \right) \leq$$

$$\underbrace{\frac{1}{3} \times \left(T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n \right)}_{\text{good split}} + \underbrace{\frac{2}{3} \left(T(n) + T(6) + n \right)}_{\text{bad split}}$$

$$T(n) \leq n + \frac{1}{3}\left(T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)\right) + \frac{2}{3} * (T(n))$$

$$T(n) * \frac{1}{3} \leq n + \frac{1}{3}\left(T(n/3) + T(2n/3)\right)$$

$$T(n) \leq 3n + T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)$$

$$T(n) \leq C * n \log n$$

$$T(n) \le 3n + T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)$$

$$T(n) \le cn\log n$$

$$T(n) \le 3n + c\frac{n}{3}\log\left(\frac{n}{3}\right) + \frac{c2n}{3}\log\left(\frac{2n}{3}\right)$$

$$\le 3n + cn\left(\log(n) - \log 3\right) + \frac{c2n}{3}\left((\log 2n) - \log 3\right)$$

$$\le n\left(3 + \frac{c}{3}\log n - \frac{c}{3}\log 3 + \frac{2c}{3}\log n - \frac{2c}{3}\log 3\right)$$

$$\le n\left(3 + c\log n - c\log 3\right)$$

$$T(n) \le cn\log n + 3n - cn\log 3$$

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

Q:- Given an unsorted array of length N, and a value K, find the K^{th} smallest element.

2	1	10	9	3	8

K = 2

$K < n$

$K \rightarrow [0, n-1]$

ans → 3

↘ Sort

$O(n \log n)$

K^{th} order Statistics

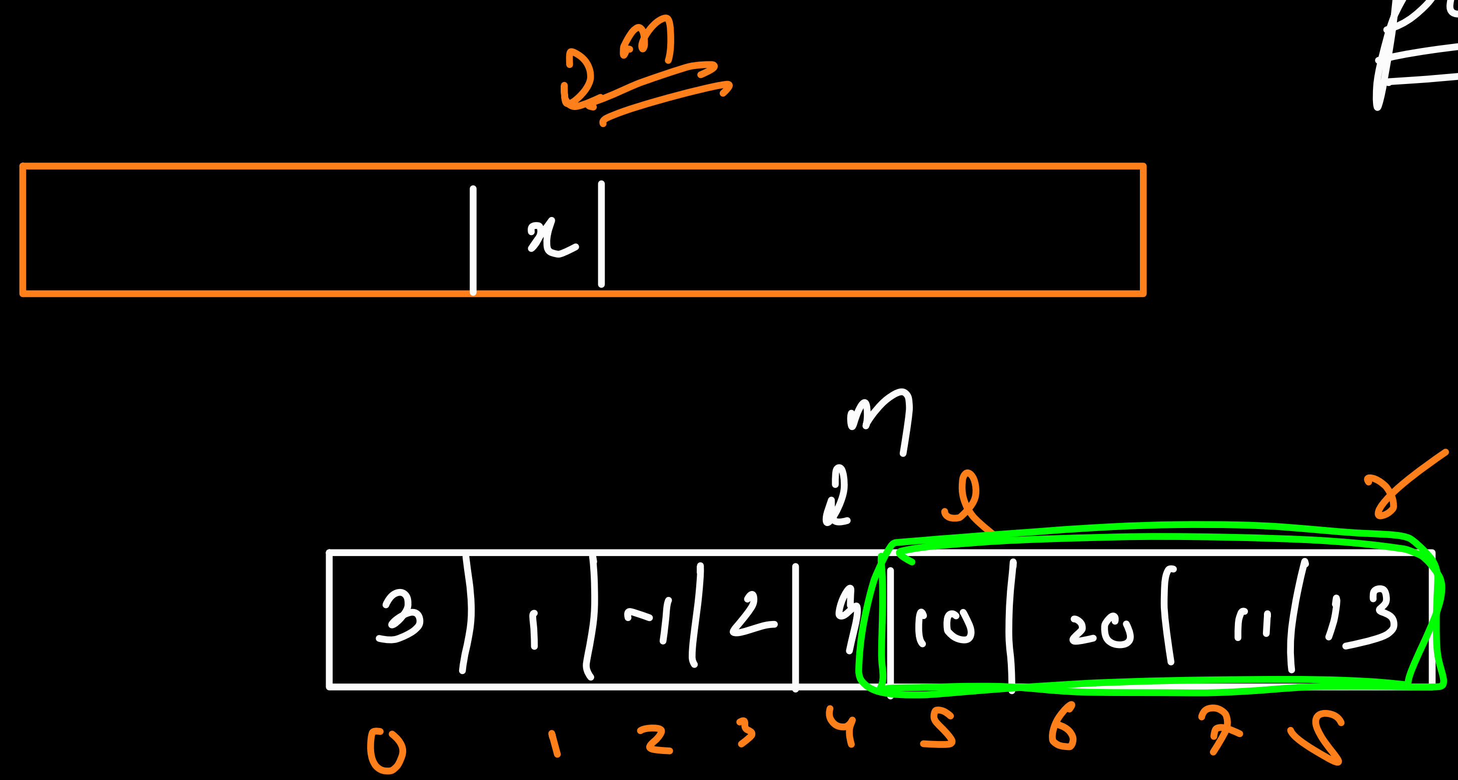
K^{th}

1	2	3	8	9	16

Quick Select

Partition → n

k = 7



if (k < m)

else

(k-5)

→ (7-5) = 2

$$\left(n + \frac{2n}{3} + \frac{4n}{9} \cdots\cdots \right)$$

$$n\left(1 + \frac{2}{3} + \frac{4}{9} + \frac{8}{27} \cdots\cdots \right)$$

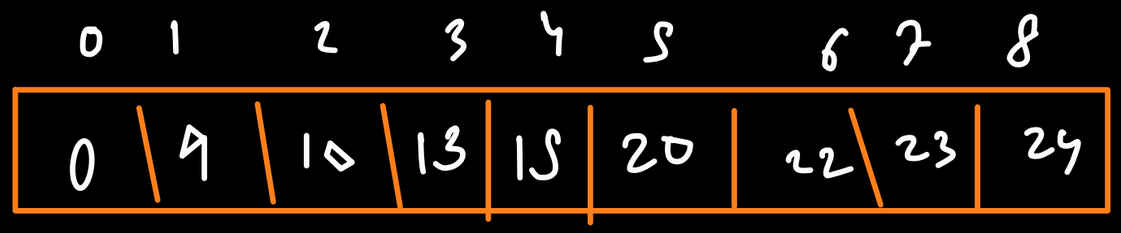
↳ G.P

$$\approx 3n$$

↳ $O(n)$

```
quickSelect (l, r, arr, k) {
    if (l > r) return;

    pi = random()
    m = partition (pi, arr, l, r);

    if (m == k)
        return arr[m];

    if (m < k)
        return quickselect (l, m-1, arr, k);

    else
        return quickSelect (m+1, r, arr, k-m-1)

}
```

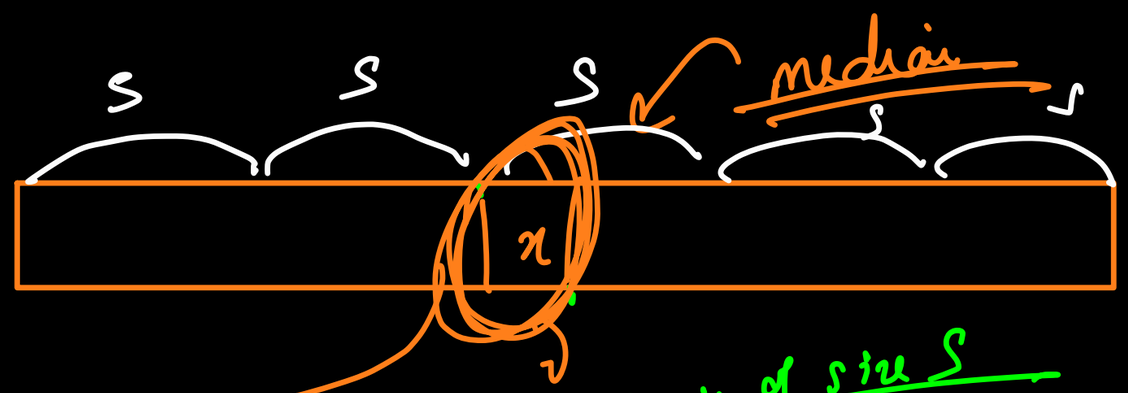
→ Randomized



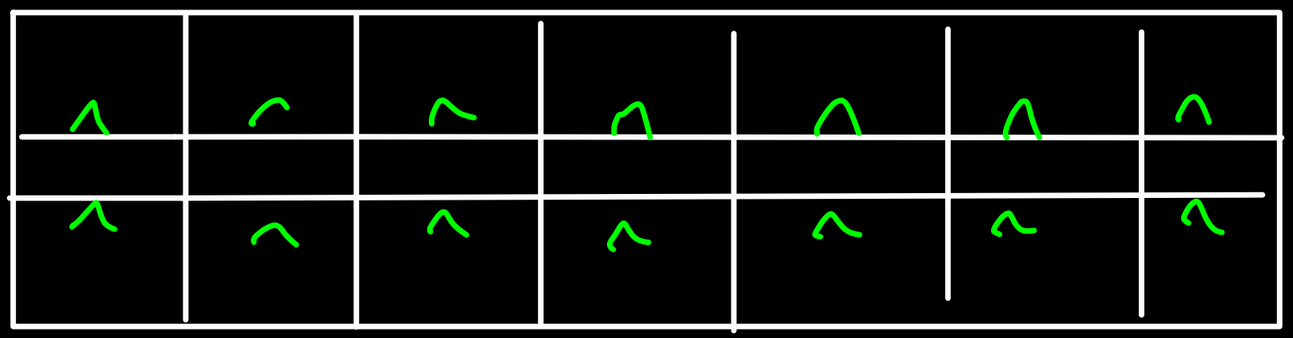
Quick Select Using BFPRT

↳ Blum Floyd Pratt Rivest Tarjan

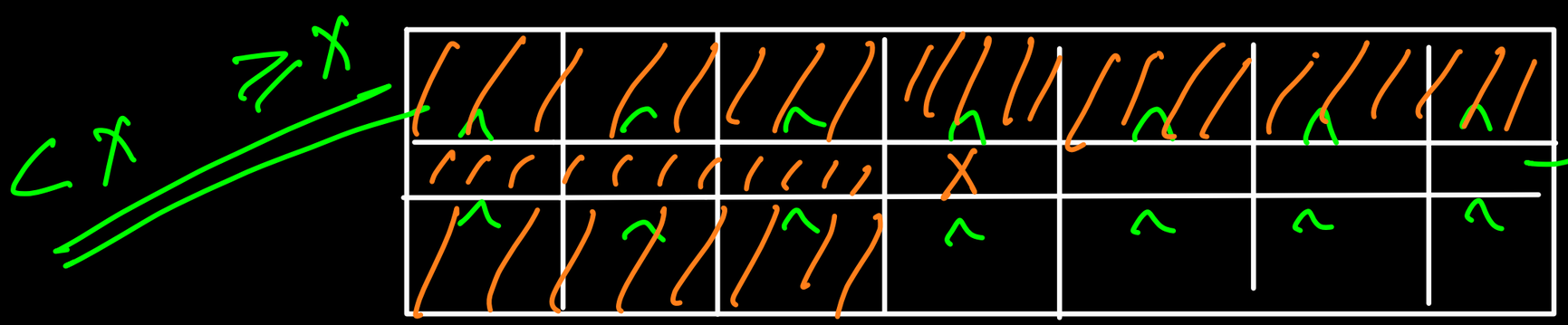
median of medians



block of size 5



apply recursively algo on the block of size 5



$n/2 + \frac{n}{2} \times \frac{2}{5} \rightarrow \frac{n}{2} + \frac{n}{5} \rightarrow \frac{7n}{10}$

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

Toproove

$$T(n) \leq c \cdot n$$

$$T(n) \leq n + \frac{cn}{5} + \frac{c7n}{10}$$

$$T(n) \leq n\left(1 + c \times \frac{9}{10}\right)$$

$T(n) \leq k \cdot n \longrightarrow T(n) \rightarrow O(n)$