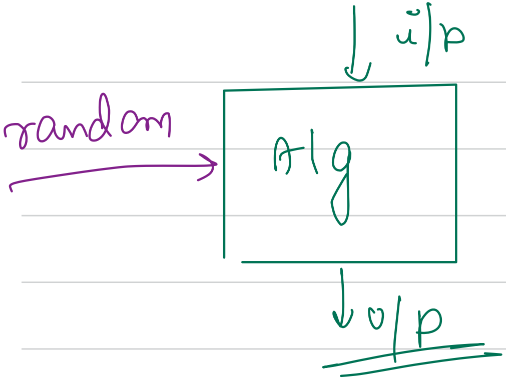
Quick Sort → Tony Hoare in 1959

Recursive Algorithm

quicksort also involves comparisons & swaps

→ partitioning
→ recursive exchange



1. deterministic algo ✓
2. randomized algo ✓

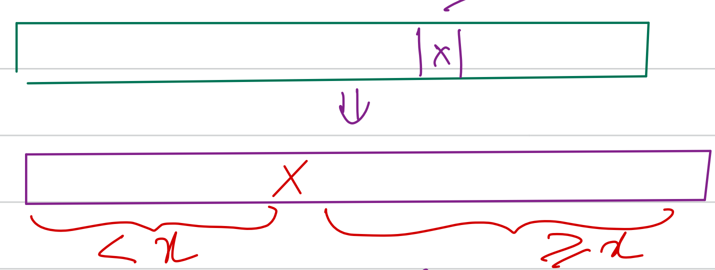
↓

quicksort

ASC

arr →

random element x → pivot element



after picking the random pivot, partition your array such that all the element $< x$ are on left side of x in any order & all elements $\geq x$ are on right side of x in any order

& then recursively apply the same procedure on left part & right part

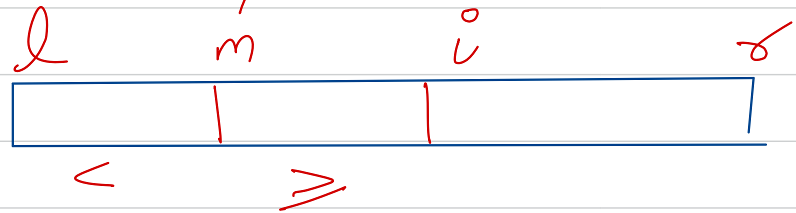
Quicksort → deterministic

→ randomized

Partitioning

Boundary

pivot x



$x = arr[random(l, r)]$ → pivot

```
m = l
for i = l .... r
    if (a[i] < x)
        swap(a[i], a[m])
        m++
```

l r m l r

~~23~~ 3, 9, 18, 32, 61, 50

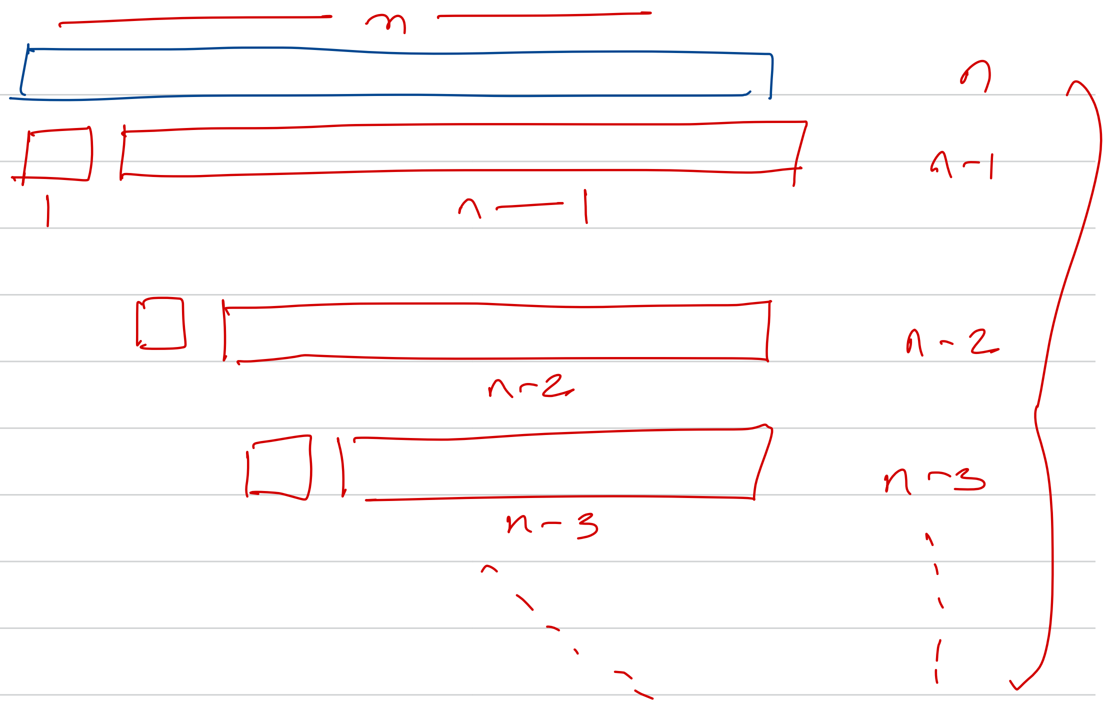
i

pivot → 32

for one iteration of partition algo → O(n)

Worst Case

Pivot → Smallest element always



The probability of hitting $O(n^2)$ is infinitely small

Worst Case time complexity $O(n^2)$

$T(n) \leftarrow$ recurrence

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

partition

$T(n)$ → no. of op. reqd to sort an array of size n by qsort

$T(n-k)$ → recurrence of qsort

So to do the analysis, we can calculate mean of time complexities taken by considering all elements as the pivot.

$$E(T(n)) = \sum_{\forall x} x \times p(T(n) = x)$$

estimated value for no. of reqd to sort the array by <u>qsort</u>

Here each is used as pivot

$[1, 1, 1, 3, 3, 5]$

$$avg = \frac{1+1+1+3+3+5}{6} = \frac{14}{6} = 2.33$$

$$avg = 1 \times p(1) + 3 \times p(3) + 5 \times p(5)$$

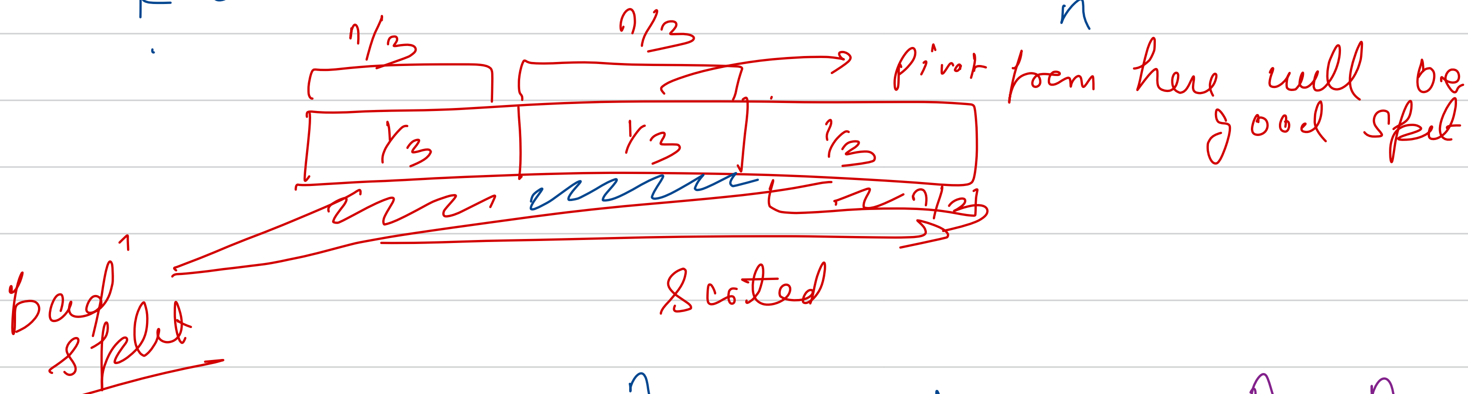
$$1 \times \frac{3}{6} + 3 \times \frac{2}{6} + 5 \times \frac{1}{6}$$

$= ?$ $\quad 14/6 = 2.33$

$$E(T(n)) = \underbrace{(n + T(1) + (T(n-1)))}_{\text{Smallest el}} \times \frac{1}{n} + \underbrace{(n + T(2) + (T(n-2)))}_{\text{2nd Smallest el}} \times \frac{1}{n}$$

— . — — — — — —

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



$$\frac{\frac{n}{3}}{n} \Rightarrow \frac{1}{3}$$

$$\frac{\frac{n}{3} + \frac{n}{3}}{n} \Rightarrow \frac{2}{3}$$

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

very small

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

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

$$T(n) - \frac{2}{3}(T(n)) \le n + \frac{1}{3}\left(T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)\right)$$

$$T(n) \times \frac{1}{3} \le n + \frac{1}{3}\left(T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right)\right)$$

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

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

Proove $T(n) \leq c \times n\log n$

c → constant

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

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

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

$$T(n) \leq 3n + cn\log n - cn\log 3 \longrightarrow \underline{O(n\log n)}$$

Worst $\rightarrow$ $T(n) = T(1) + T(n-1) + O(n)$

Avg $\rightarrow$ $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n)$

Best $\rightarrow$ $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + O(n)$

Order statistics

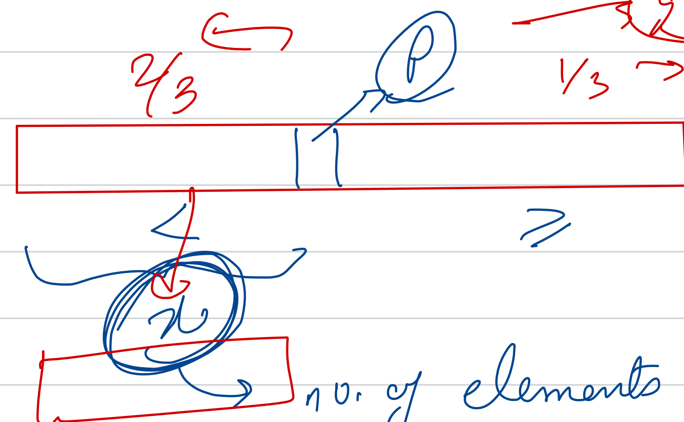
$\downarrow$

$O(n)$ time

quick select

Given an array find the K^{th} smallest element.

good split



no. of elements less than p

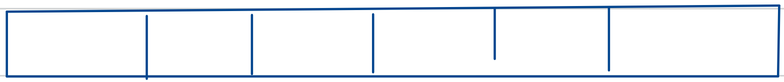
$x > K$

$$\left(n + \frac{2}{3}n + \frac{4}{9}n \cdots\cdots\right) \approx O(n) \rightarrow \text{randomization}$$

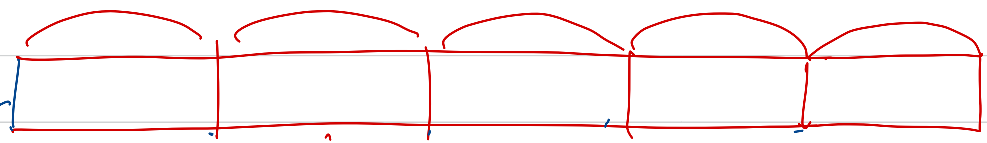
$\downarrow$ $3n$

Deterministic algo → median of median

arr →



→ S blocks



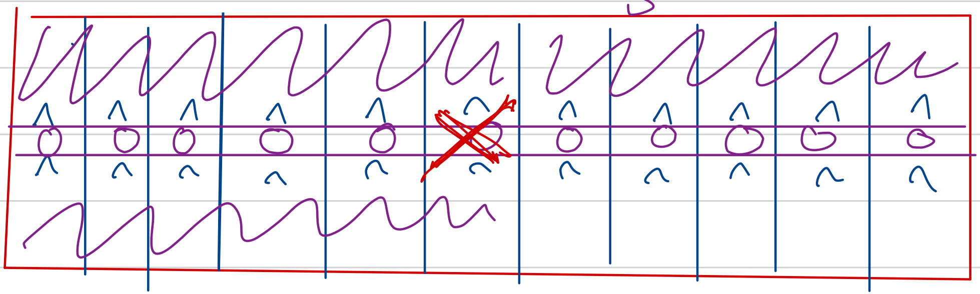
size S

all blocks of size S

median of medians → order statistic

$\frac{n}{2} + \frac{2}{5} \times \frac{n}{2}$

Block median →



Total blocks → $\frac{n}{5}$

n

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

worst case

Individually sort the 5 size block

$$T(n) = \underbrace{n + T\left(\frac{n}{5}\right)}_{\text{median}} + \underbrace{T\left(\frac{7n}{10}\right)}_{\text{recursive}}$$

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

$$T(n) \le c \cdot n$$ $c \rightarrow$ const

$$T(n) \le n + \frac{cn}{5} + \frac{c7n}{10}$$ $\frac{2+7}{10}$

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

$$T(n) \le K \times n$$

$$T(n) \rightarrow O(n)$$

stability → No

Inplace → yes

Space complex → O(log n)

Cache friendly → yes → locality of ref