ECB19035

Quick Sort Analysis

08:00

Steps for quick sort:

09:00 -> partition the array & set a pivot.

-> Recursively call left & right part of pivot to sort itself

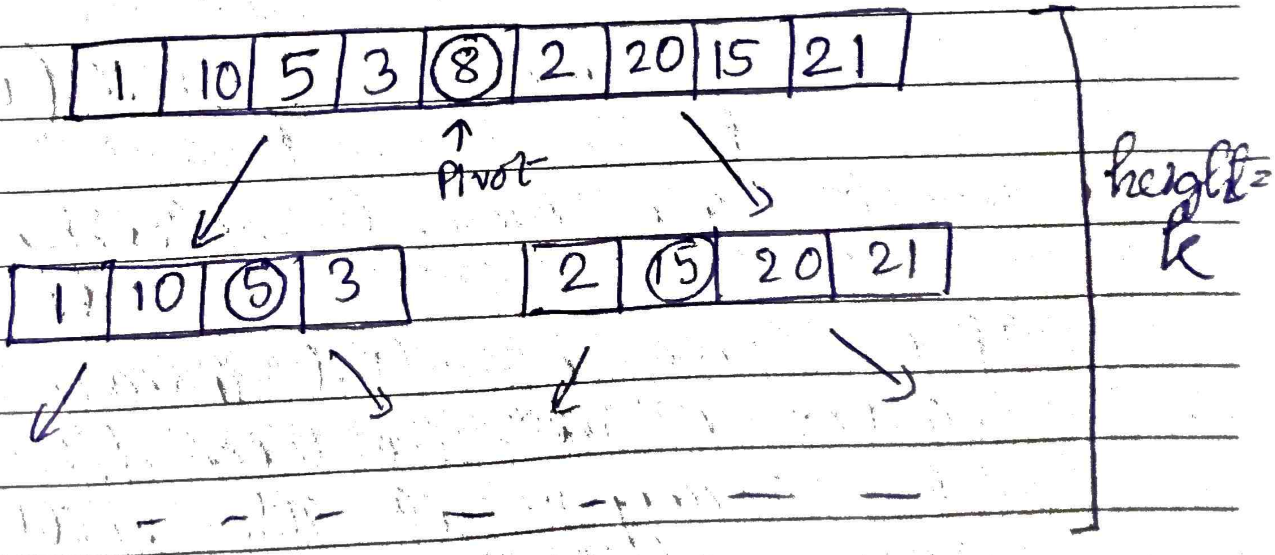
10:00

11:00 • Best Case Time Complexity = O(nlogn)

12:00 If we assume that the pivot is

13:00 always the median of the array then,

14:00



15:00

16:00

17:00

18:00

If the height of recursive tree

19:00 is k & the array goes on to decrease its range by half, so

Eve. ~~have~~ we have to divide the array $n/2^k$ times & until only one element is left.

So,

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\boxed{k = \log_2 n}$$

So, no. of times we do ~~partition~~ compare & pivot is = n
& we have $\log_2 n$ times division of array.

∴ Time Complexity = $O(n \log n)$

- Worst Case Time Complexity = $O(n^2)$

If we have a sorted array & the pivot is at the beginning of the array or at the end of the array, then,

| ① | 2 | 3 | 4 | n times

→ | ② | 3 | 4 | n-1 "

→ | ③ | 4 | n-2 "

→ | 4 | n-3

We know,

$$n + (n-1) + (n-2) + (n-3) + \cdots$$

$$> \frac{n(n+1)}{2}$$

$$= \frac{n^2+n}{2}$$

Here, the Time Complexity comes out to be $\underline{O(n^2)}$.

- To Optimise the Worst Case of <u>Quick Sort</u>
  - – Select Middle element as the pivot.
  - – Select Random element as pivot.

August 2015

24 Monday

(236 - 129) Wk 35

August ◂

Su	Mo	Tu	We	Th	Fr	Sa
30	31	•	•	•	•	
2	3	4	5	6	7	
9	10	11	12	13	14	
16	17	18	19	20	21	
23	24	25	26	27	28	

Space Complexity

08:00

09:00

Quick Sort is a recursive algorithm so it doesn't require any extra space, but it will use Stack.

10:00

- In Worst case, quick sort may take Stack size of n

11:00

12:00

- In Best case, quick sort may take Stack size of log n

13:00

So, it can vary from

14:00

log n → n

15:00

16:00

17:00

18:00

19:00

Eve.