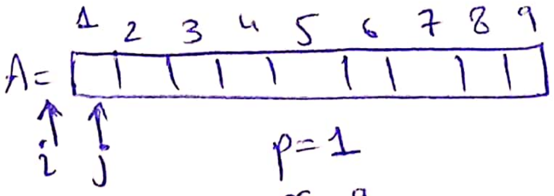
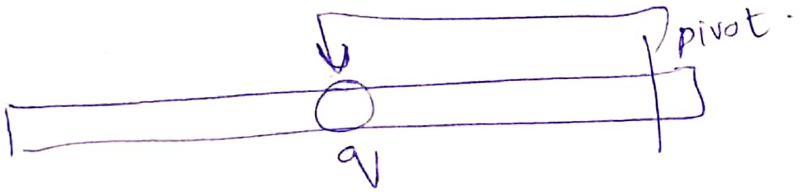
```
Partition (A, $\overset{l}{p}$, $\overset{h}{r}$)        ③
{
    x = A[r]
    i = p-1
    for (j=p to r-1)
        if (A[j] ≤ x)
            i = i+1
            exchange A[i] with A[j]
    exchange A[i+1] with A[r]
    return i+1
}
```



p=1
r=9
pivot=x=A[9]
i=p-1=0
j=p=1

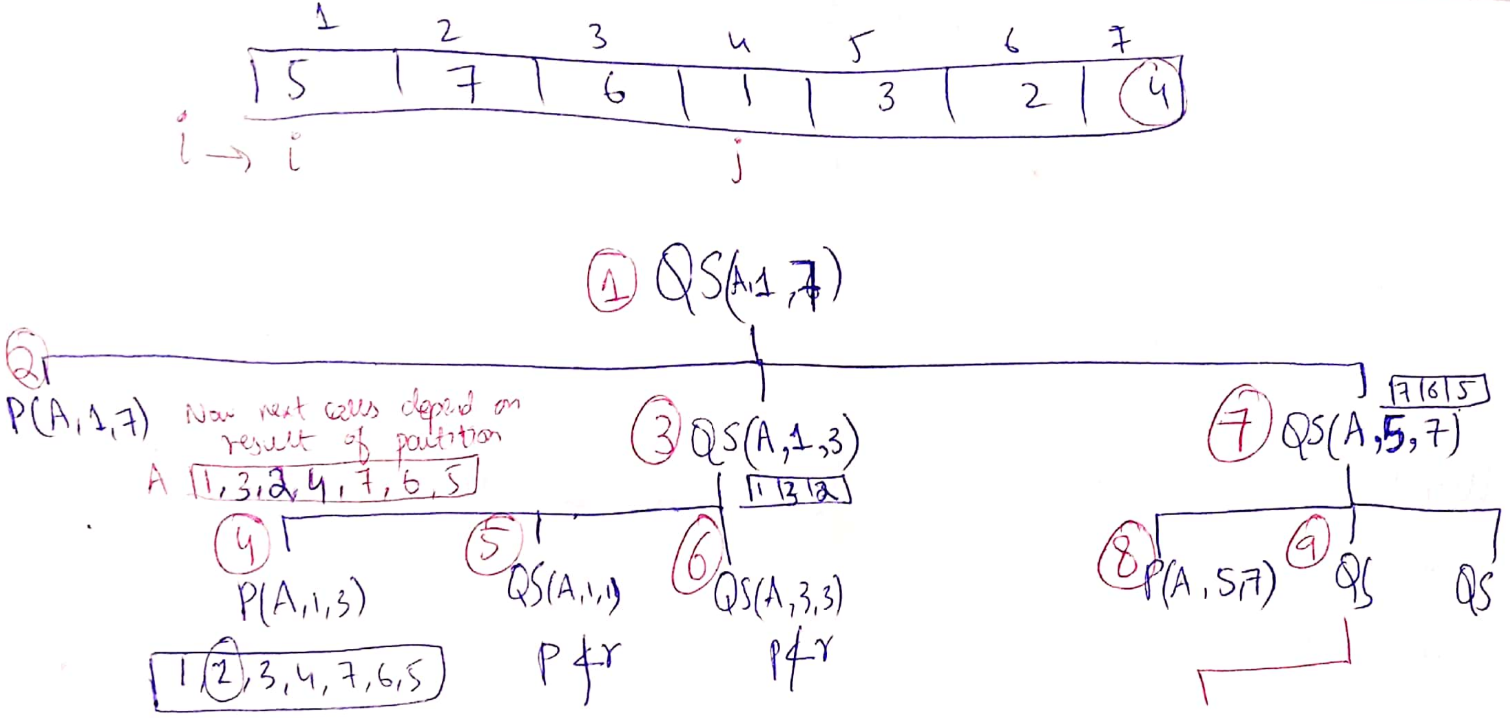
```
④ Quicksort (A, p, r)
      if (p<r)
① {     q = partition (A, p, r)
②       Quicksort (A, p, q-1)
③       Quicksort (A, q+1, r)
   }
  }
```

line 1



partition can also take first element as pivot.

Dry Run (Quick-sort)



Space complexity:

In-place Algorithm.

⑥

Time Complexity:

⑦

Time To sort n elements $= T(n)$

Palition — $O(n)$

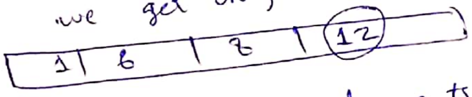
Best case analysis:

partitioning gives mid point every time as pivot element

$$T(n) = 2T\left(\frac{n}{2}\right) + cn$$

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

Worst case analysis:



we get only one part which has all the n-1 elements.

zero elements on one side & $n-1$ elements on the other side.

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

$$T(n) = cn + T(n-1) \longrightarrow ①$$

Base case $T(1) = 0 = \Theta(1)$

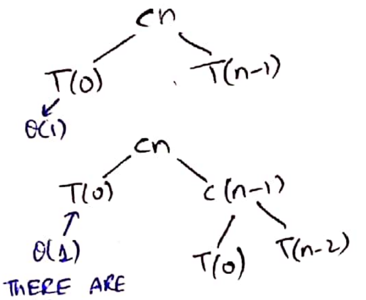
Solve by iterative method.

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

$$= T(n-3) + c(n-2) + c(n-1) + cn$$

$$\vdots$$

$$= T(n-k) + (n-k-1) + \cdots + cn$$



SO THERE ARE n Θ(1)s, So

height of tree is n.

$n-1$

$T(n) = \Theta(n) + \Theta(n^2)$ $\Theta\left(\sum_{k=1}^{n} ck\right) = \Theta(n^2)$

$n-k=1$ $\boxed{k=n-1}$

$= T(1) +$

→ Ascending order

((1 2 3 4) ⑤) ⑥)

→ Descending order

⑥ 5 4 3 2 1

(① (5 4 3 2 6))

→ All elements same

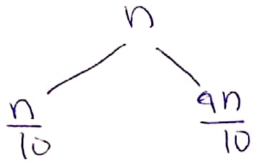
((5 5 5 5) ⑤)

Average Case Analysis :

$\boxed{\frac{1}{10}, \frac{9}{10} \text{ split}}$ Is this luck or unlucky?

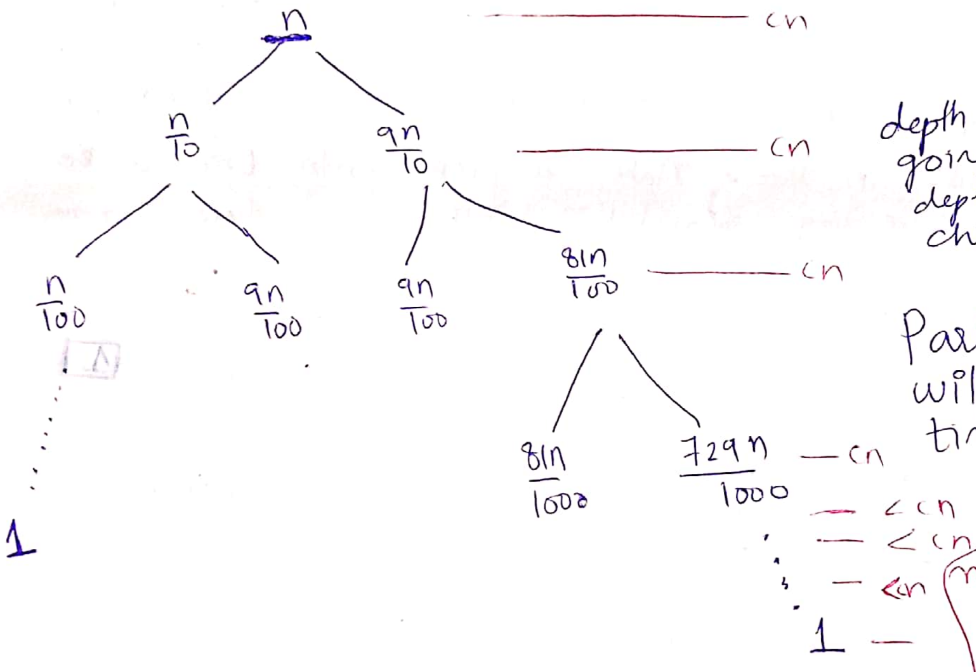
Size of two sub-problems is in the ratio of 1:9

So if we have n elements, we get



If we make 10 parts of input, 9 are in one sub-array & 1 is in 2nd sub-array. Now this seems pretty unbalanced partitioning.

$$T(n) = T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + cn$$



depth of left side is going to be less than depth of right side child of tree.

Partitioning procedure will take O(n) time at every level

cuz elements are not n now so work done is decreasing.

Size of sub problem in Right child.

$n, \frac{n}{10/9}, \frac{n}{(10/9)^2}$ . . . . . 1 Still logarithmic but what base??

not decreasing by half, but by 10/9.

when decrease by half, we take $\log_2$

$\boxed{\log_{10/9} n}$

In left child $\log_{10}(n)$

How logs of diff bases are asymptotically related?

$\log_{10/9}(n) = \Theta(\log_2 n)$

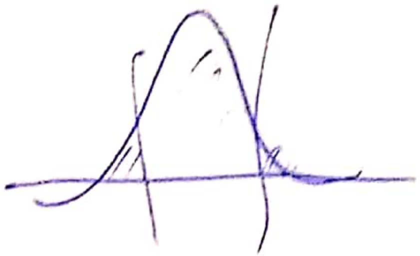
$\log_{10}(n) = \Theta(\log_2 n)$

So levels are $\Theta(\log_2 n)$

So, total work $= \Theta(n \log n)$

So, $\frac{9n}{10}, \frac{n}{10}$ split is asymptotically as good as $\frac{n}{2}, \frac{n}{2}$ split

So



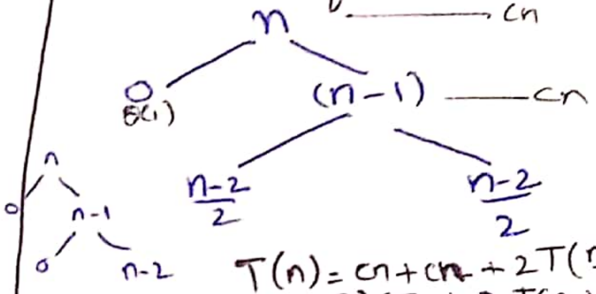
$$T(n) \le cn \log_{10/9} n$$

We can also see lower bound by the branch which finishes earlier

$$cn \log_{10} n \le T(n) \le cn \log_{10/9} n$$

So, $T(n) = \Theta(n \log n)$

When I sometimes get a good split & then bad then good then bad, quick sort will work in nlogn. It only works in n^2 where all splits are bad which is the case of sorted array.



$$T(n) = cn + cn + 2T\left(\frac{n-2}{2}\right)$$
$$= 2cn + 2T\left(\frac{n-2}{2}\right)$$
$$\le O(n) + 2T(n/2) = O(n\log n)$$

tree method to make a guess