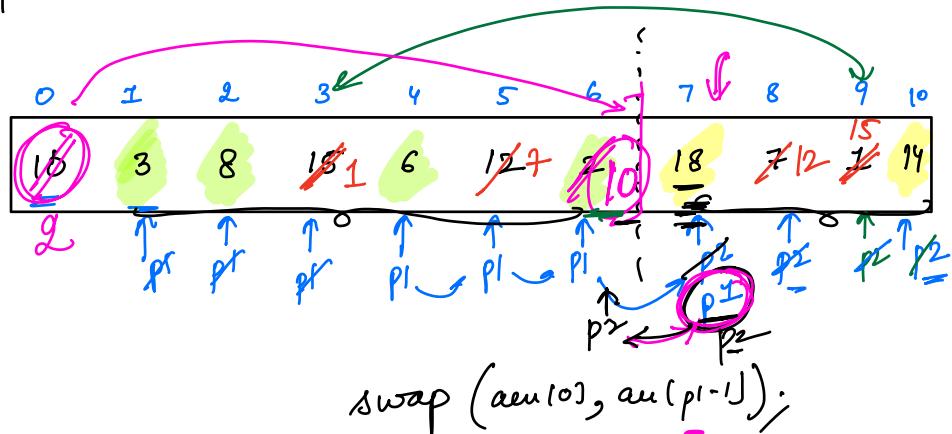
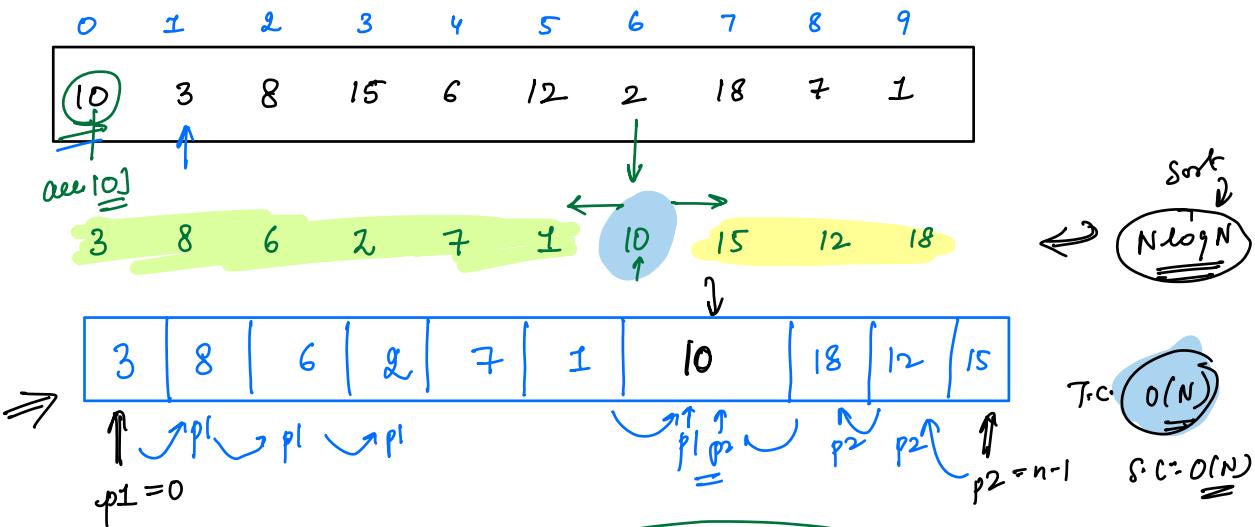


Q Given an array of N elements. Rearrange the array such that

$\text{arr}[0]$ should go to its sorted position &

All the elements smaller than $\text{arr}[0]$ should

go on left & all elements greater should go on right.



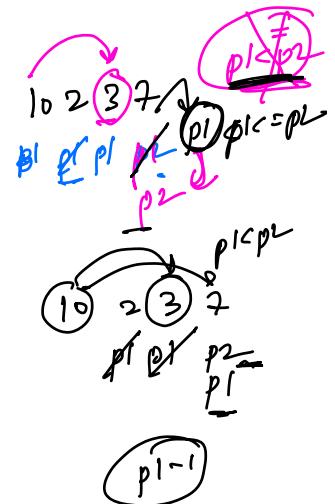
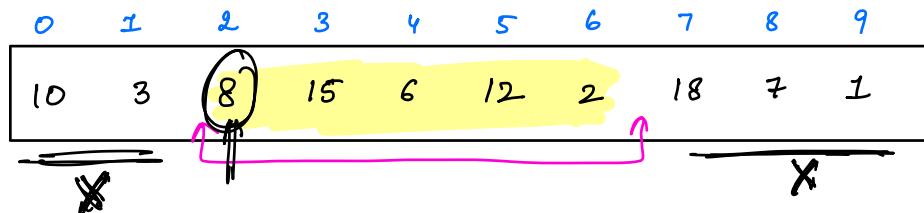
```

arr p1=1, p2=n-1;
while ( p1 <= p2)
{
    if ( arr[p1] <= arr[0])
    {
        p1++;
    }
    else if ( arr[p2] > arr[0])
    {
        p2--;
    }
    else
    {
        swap( arr[p1], arr[p2]);
        p1++; p2--;
    }
}

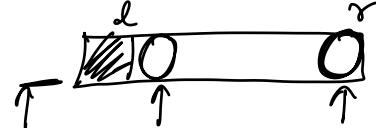
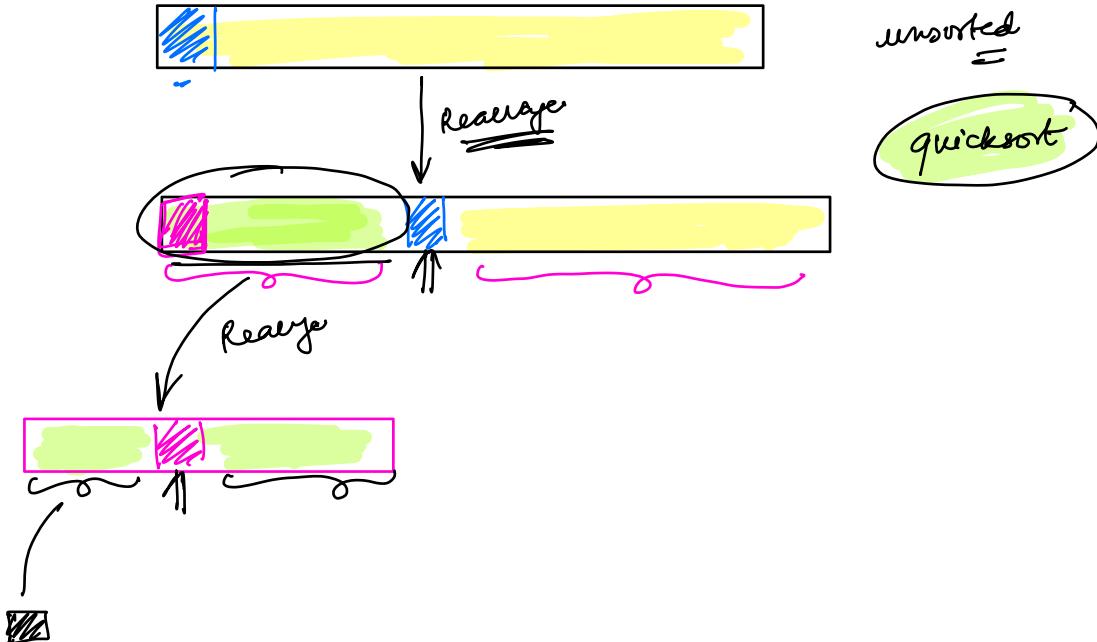
```

swap(arr[0], arr[p1-1]);

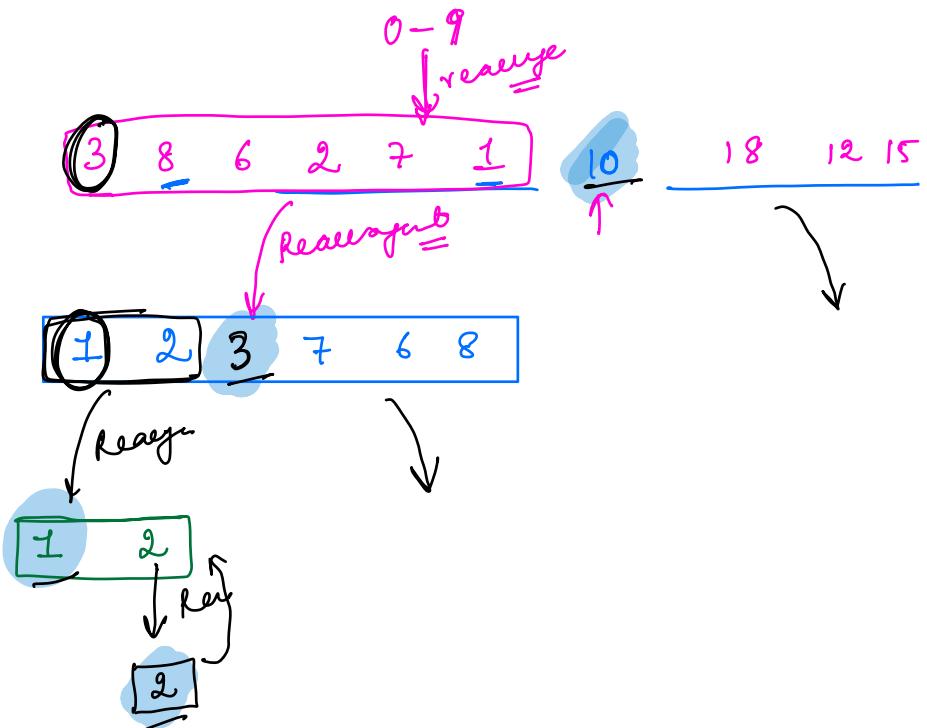
 (l, r) \rightarrow rearrange the range
 $|$
 $arr[l:r]$



int rearrange (arr, l, r)
 {
 rand point
~~int p1 = l+1, p2 = r-1;~~
 while (p1 <= p2)
 {
 if (arr[p1] <= arr[0])
 {
 p1++;
 }
 else if (arr[p2] > arr[0])
 {
 p2--;
 }
 else {
 swap(arr[p1], arr[p2]);
 p1++; p2--;
 }
 }
 swap(arr[0], arr[p1]);
 return p1-1;
}

0	1	2	3	4	5	6	7	8	9
10	3	8	15	6	12	2	18	7	1



void quicksort (int au[], int l, int r)

{

 if ($l \geq r$) return; $l \geq r$

 int ind = partition(au, l , r)

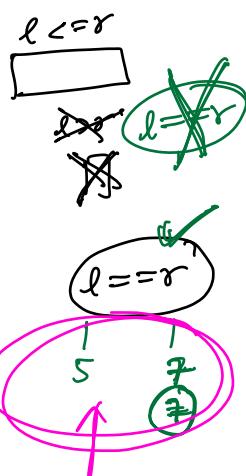
$s - c$

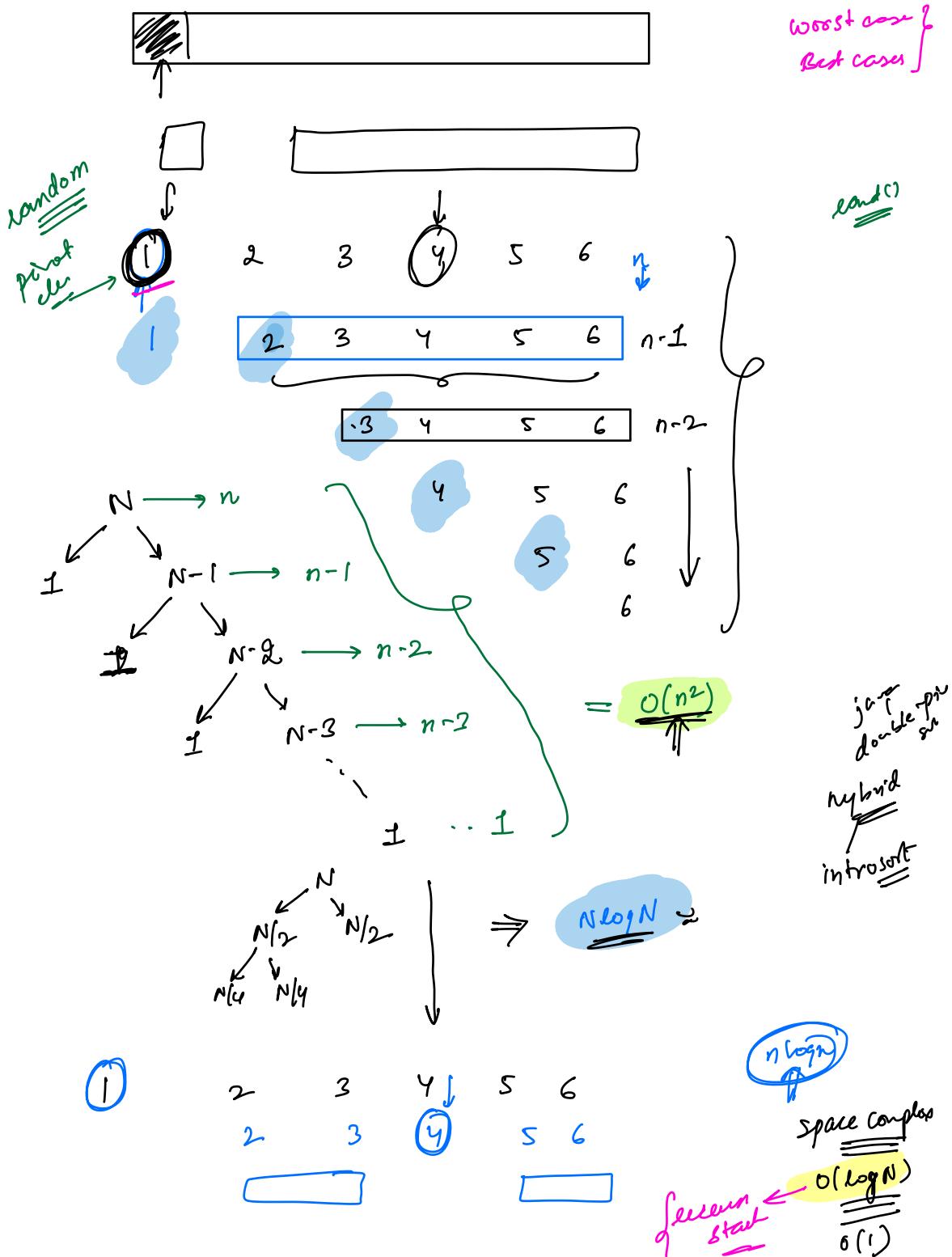
 quicksort(au, l , $index-1$);

 quicksort(au, $index+1$, r);

$8,7$

}





Given an array of N elements. Find sum of (Max-Min) for every subsequence.

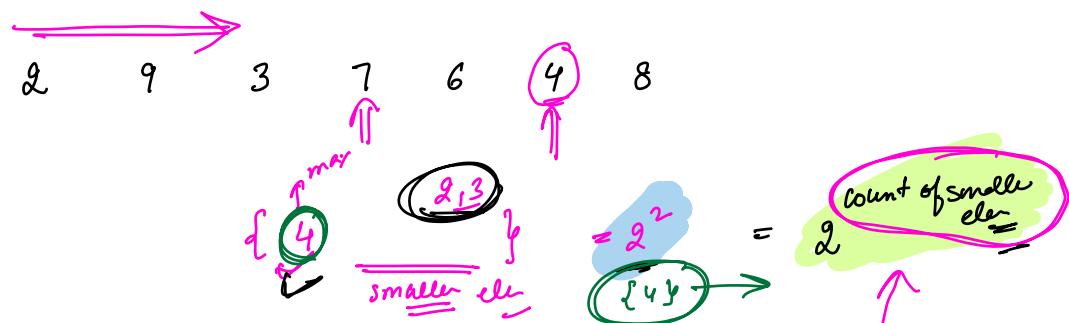
$\{3, 1, -4\}$	$\Rightarrow \{-4, 3\}$
max	min
3	3
1	1
-4	-4
3	1
1	-4
3	-4
3	-4
<u>10</u>	<u>-11</u>
<u>$10 - (-11)$</u>	<u>21</u>

BoF → generate all subsequences → 2^n

$$\begin{aligned}
 & \sum_{\text{subseq}} \text{max}_{\text{subseq}} - \text{min}_{\text{subseq}} \\
 &= \sum \text{max}_s - \sum \text{min}_s
 \end{aligned}$$

When will any element contribute?
↓
Max of a subsequence
↓
How many subsequences
any element is max
Contributiv = count × elemt

Sum of all contrib

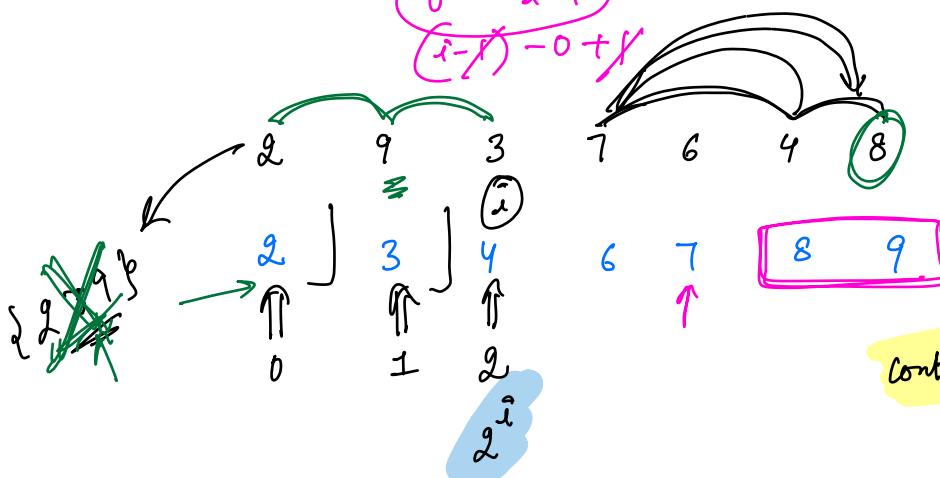


$$\text{element} = \underset{\approx \max}{2} \times \underset{\text{count of small el}}{2} \times \underset{\approx \text{element}}{2}$$

$N \times N = O(N^2)$



$$(0 - i-1) \\ (i-1) - 0 + 1$$



$$\{8\} \times 2$$

$N^2 \downarrow$
 $N \times N \downarrow$

after sorting

$$\text{cont} = \underset{\approx \text{all}[i]}{2^i} \times \underset{\approx \text{all}[i]}{\text{all}[i]}$$

subsequence

pick any ele in
degree order

$$\sum_{i=0}^n 2^i \times \text{all}[i]$$

$$\sum_{i=0}^n = ?$$

$$\sum_{i=0}^n \downarrow \text{count of large el}$$

2 9 3 7 6 4 8

$$\sum_{min} = 2^{n-i-1} \times au[i]$$

$$au[i] = 2^i \times au[1] - 2^{n-i-1} \times au[1]$$

$$f = au[i] \times (2^j - 2^{n-i-1})$$

$\Theta(n \log n)$

$$345 / 10 = 5$$

i^{th} digit \rightarrow $\left(\frac{N}{10^i} \right) \% 10$

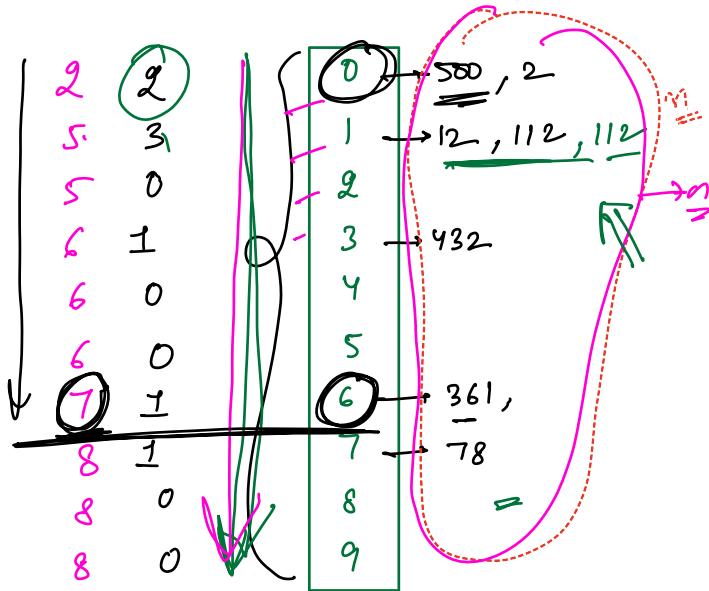
$$500 / 10 = 0$$

$$432 / 10 = 4$$

$$361 / 10 = 1$$

0-9

$n \rightarrow \frac{\text{Radix}}{(N/10^i) \% 10}$



T.C: $n + \underline{\underline{10}}$
 $O(n+10) \approx O(\underline{\underline{n}})$
S.C: $O(n)$

