Q.1 Given an Array of size N. find the K^{th} minimum element in the Array. $K < \log_2 N$

A : { 1, 5, -1, 2, 10, 3 }

$K = 3 \Rightarrow 2$

$K = 5 \Rightarrow 5$

$K = 1 \Rightarrow -1$

Brute force

1. sort (Arr)
2. return Arr[K-1]

A : { 1, 5, -1, 2, 10, 3 }

$\downarrow$ sort

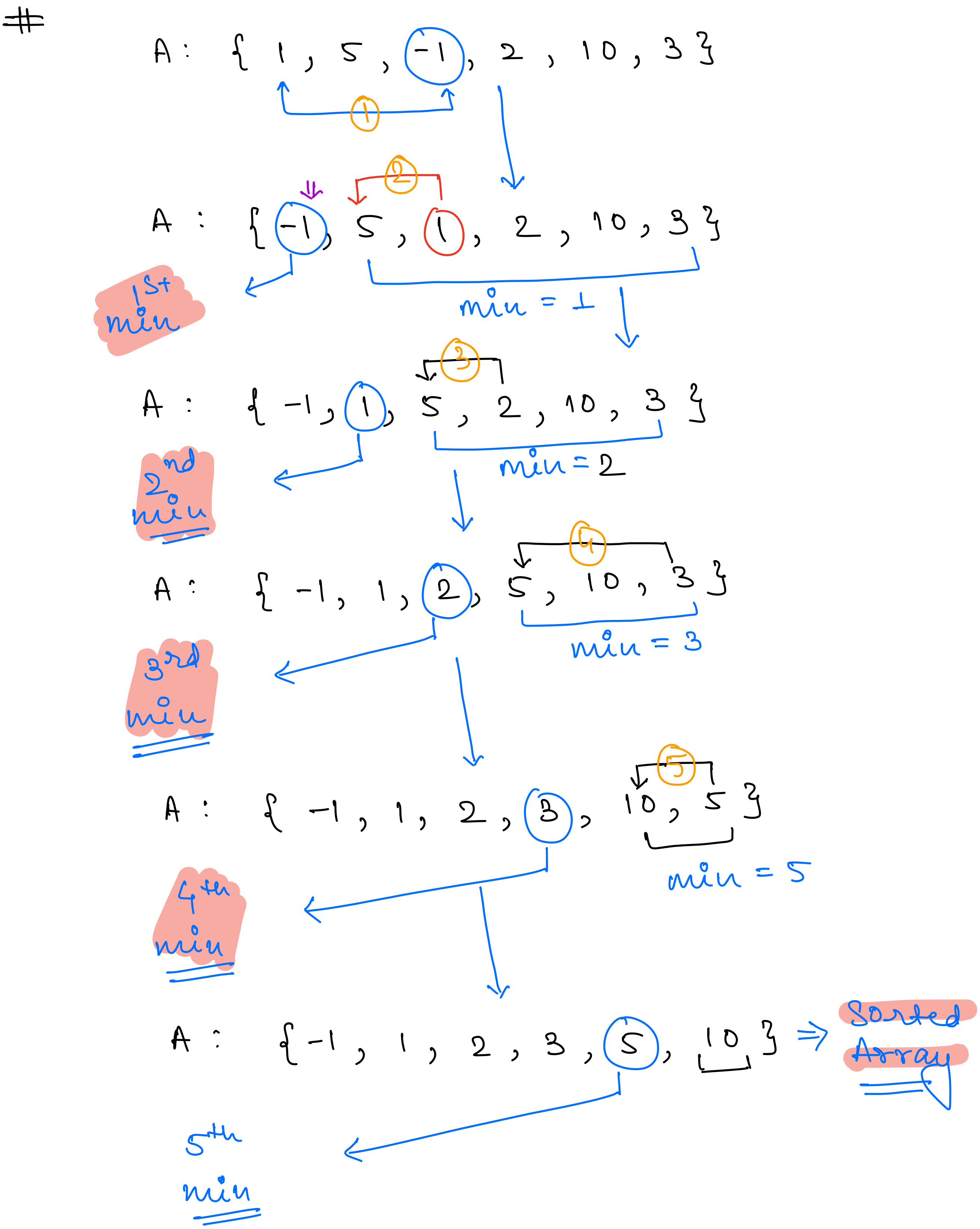
$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ \{-1, & 1, & 2, & 3, & 5, & 10\} \end{array}$$

$K = 3 \Rightarrow A[2]$

TC : $O(N \log N)$

SC : Depends on sorting algo.

#



A : { 1 , 5 , -1 , 2 , 10 , 3 }

1

A : { -1 , 5 , 1 , 2 , 10 , 3 }

2

1st min

min = 1

A : { -1 , 1 , 5 , 2 , 10 , 3 }

3

2nd min

min = 2

A : { -1 , 1 , 2 , 5 , 10 , 3 }

4

3rd min

min = 3

A : { -1 , 1 , 2 , 3 , 10 , 5 }

5

4th min

min = 5

A : { -1 , 1 , 2 , 3 , 5 , 10 } ⇒ Sorted Array

5th min

TC of finding the Min(Arr) = $O(N)$

TC: $O(KN) < O(N\log N)$

SC : $O(1)$

Selection Sort

Selecting the minimum element & placing this element at its right position is Selection Sort.

TC : $O(N^2)$

```
for ( i = 0; i < N; i++ ) {
        min = A[i]
        min_index = i;
        for ( j = i+1; j < N; j++ ) {
                if ( A[j] < min ) {
                        min = A[j];
                        min_index = j;
                }
        }
        swap( A[i] , A[min_index] );
}
```

$$A : \{ \underset{\overset{\uparrow}{i=0}}{\overset{0}{1}}, \overset{1}{5}, \overset{2}{\boxed{-1}}, \overset{3}{2}, \overset{4}{10}, \underset{\uparrow}{\overset{5}{3}} \}$$

$$\left.\begin{array}{l} min = \cancel{1}\ -1 \\ m\_i = \cancel{0}\ \ 2 \end{array}\right) \Rightarrow swap(A[0], A[2])$$

$$A : \{ \overset{0}{-1}, \underset{\overset{\uparrow}{i}}{\overset{1}{5}}, \underbrace{\overset{2}{1}, \overset{3}{2}, \overset{4}{10}, \overset{5}{3}} \}$$

$$min = \cancel{5}\ 1$$

$$m\_i = \cancel{0}\ \ 2$$

$$A : \{ \overset{0}{-1}, \overset{1}{1}, \underset{\overset{\uparrow}{i}}{\overset{2}{5}}, \underset{\overset{\uparrow}{j}}{\overset{3}{\boxed{2}}}, \overset{4}{10}, \overset{5}{3} \}$$

$$min = \cancel{5}\ 2$$

$$m\_i = \cancel{2}\ 3$$

$$A : \{ -1, 1, 2, \underset{\overset{\uparrow}{i}}{5}, 10, 3 \}$$

$$min = \cancel{5}\ 3$$

$$m\_i = \cancel{3}\ 5$$

$$A : \{ -1, 1, 2, 3, \underset{\overset{\uparrow}{i}}{10}, \underset{\overset{\uparrow}{j}}{5} \}$$

$$min = \cancel{10}\ 5$$

$$m\_i = \cancel{4}\ 5$$

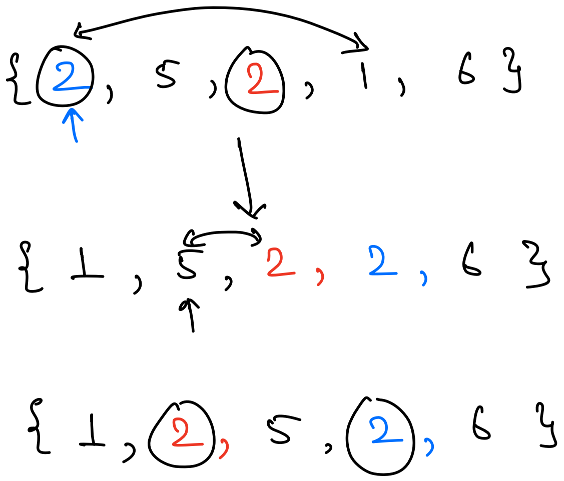
A : {-1, 1, 2, 3, 5, 10}

TC : O(N^2)

Quiz

Inplace? ⇒ YES : SC = O(1)

Stable ⇒ NO



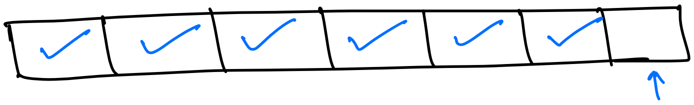
{2, 5, 2, 1, 6}

{1, 5, 2, 2, 6}

{1, 2, 5, 2, 6}

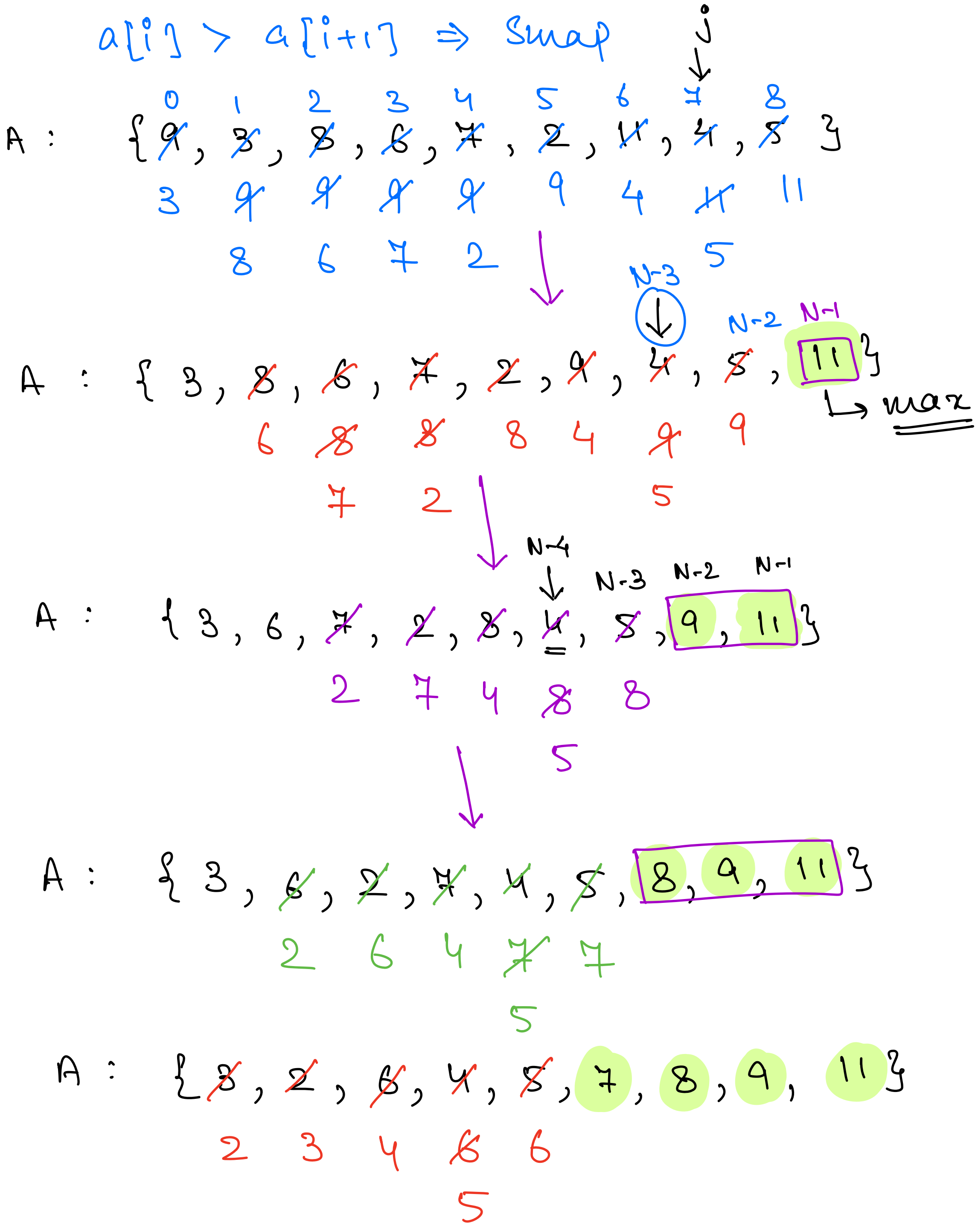
Quiz Max. no. of swaps in selection sort :-

= N-1



Constraint :- Swapping is allowed only b/w consecutive elements.

a[i] > a[i+1] ⇒ Swap



j
0 1 2 3 4 5 6 7 8
A : { 9, 3, 8, 6, 7, 2, 11, 4, 5 }
3 9 9 9 9 9 4 11 11
8 6 7 2 5
N-3
N-2
N-1
A : { 3, 8, 6, 7, 2, 9, 4, 5, 11 }
6 8 8 8 4 9 9
7 2 5
max
N-4
N-3
N-2
N-1
A : { 3, 6, 7, 2, 8, 4, 5, 9, 11 }
2 7 4 8 8
5
A : { 3, 6, 2, 7, 4, 5, 8, 9, 11 }
2 6 4 7 7
5
A : { 3, 2, 6, 4, 5, 7, 8, 9, 11 }
2 3 4 6 6
5

A : { 2, 3, 4, 5, 6, 7, 8, 9, 11 }

⇒ Sorted

Keep on taking the Max. element at the end of the Array.

⇒ Bubble Sort

Code

```
for( i=0; i< N; i++){
    for( j=0; j< N-1; j++){
        if( A[j] > A[j+1]){
            swap(A[j], A[j+1]);
        }
    }
}
```

$j <= N-i-2$

$j < N-i-1$

Swap

$i = 0 \Rightarrow j \in [0, \overset{N-0-2}{N-2}] \Rightarrow N-1$

$+$

$i = 1 \Rightarrow j \in [0, \overset{N-1-2}{N-3}] \Rightarrow N-2$

$+$

$i = 2 \Rightarrow j \in [0, \overset{N-2-2}{N-4}] \Rightarrow N-3$

$+ \dots + 1 + 0$

$i \Rightarrow j \in [0, N-i-2]$

$0 =< j <= N-i-2$

* If at any iteration, swap count is zero then it means Array has already become sorted.

TC : $O(N^2)$

Quiz Max. no. of swaps in Bubble Sort.

swaps = $1 + 2 + 3 + \text{------} + (N-1)$

$= \boxed{\frac{N(N-1)}{2}}$

Inplace ? YES | SC: O(1)

Stable ? YES

A : { $\cancel{2}$, $\cancel{1}$, $\cancel{5}$, $\cancel{2}$ }

1 2 2 5

A : { 1 , 2 , 2 , 5 }

Q. Given 2 sorted arrays of size N & M. Merge these 2 sorted arrays into 1 sorted array.

Amazon

A: { 2, 5, 7, 12, 20, 24, 29 }

B : { 6, 9, 10, 14, 18, 19 }

C: { 2, 5, 6, 7, 9, 10, 12, 14, 18, 19, 20, 24, 29 }

Approach#1

⇒ int C[N+M]

⇒ Put all the elements of A & B into C. ↳ N+M

⇒ sort (C) ⇒ O((N+M) log(N+M))

TC: O((N+M) log(N+M))

SC: O(N+M)

Approach#2

A: {2, 5, 7, 12, 20, 24, 29} ⇒ Sorted

↑ i (at 20)

B: {6, 9, 10, 14, 18, 19} ⇒ Sorted.

↑ j (after 19)

C: $\overset{0}{2}, \overset{1}{5}, \overset{2}{6}, 7, 9, 10, 12, 14, 18, 19, 20, 24, 29$}

(N+M)

C[0] = min(A[0], B[0])

C[K] = min(A[i], B[j])

```
merge (int A[], int M, int B[], int N){
    int  c [N+M]
    i=0  // Array A
    j=0  // Array B
    K=0  // Array C
    while( i< M  && j<N){

        if(A[i] < B[j]){
            C[K] = A[i];
            i++
            K++
        }
        else{
            C[K] = B[j]
            j++
            K++
        }
    }
    while( i< M) {
        C[K] = A[i];
        i++
        K++
    }
    while(j < N) {
        C[K] = B[j];
        j++;
        K++;
    }
    return c;
}
```

TC : $O(N+M)$

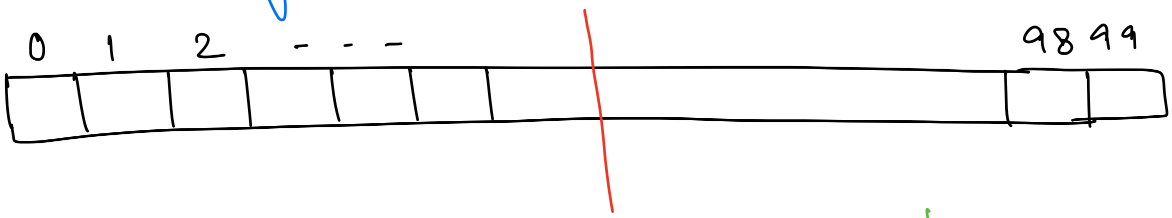
SC : $O(N+M)$ (Output Array)

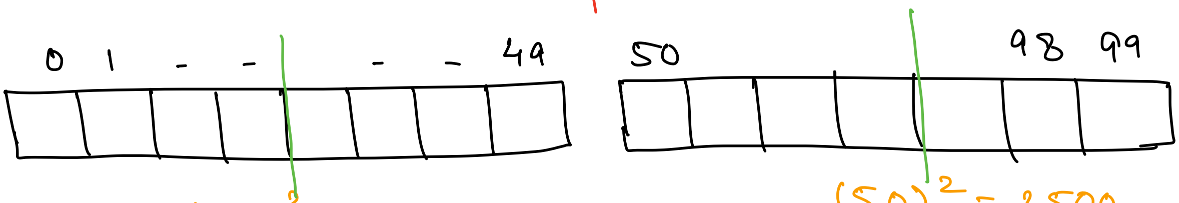
Old machine

Task :- Sort an Array of size N = 100.

Bubble Sort
Selection Sort $\Big\} O(N^2)$

# of iterations $= (100)^2 = \underline{\underline{10000}}$



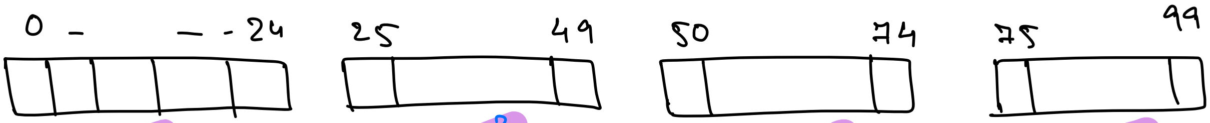


$(50)^2 = 2500$ $(50)^2 = 2500$

100

merge

# of iterations $= 2500 + 2500 + 100$
$= 5100$



$(25)^2$ $(25)^2$ $(25)^2$ $(25)^2$

25+25 25+25

50+50

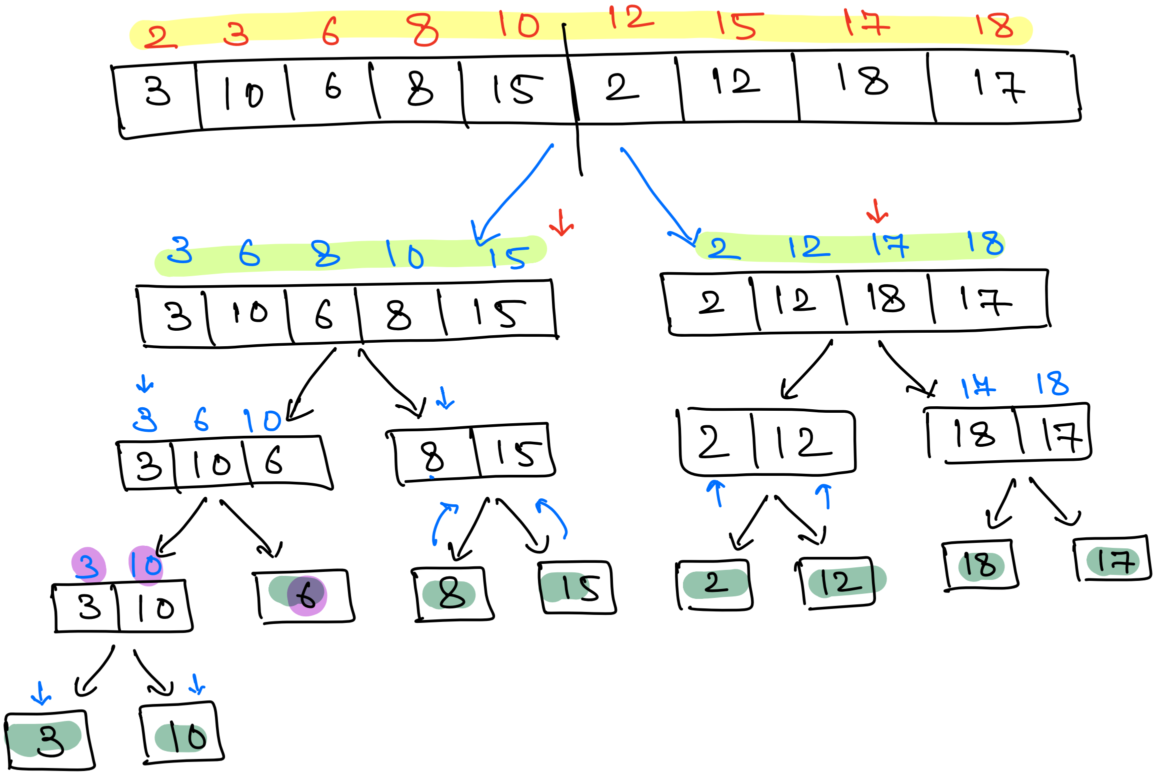
$\# \text{ of Iterations} = (25)^2 * 4 + 25*4 + 50+50$

$= 2500 + 100 + 100$

$= \underline{\underline{2700}}$

10K ⟶ 5K ⟶ 2.5K ⟶ - - -

MERGE SORT (Divide & Conquer)



Code :-

Assumption :- mergeSort (A, l, r) sorts the Array from l to r.

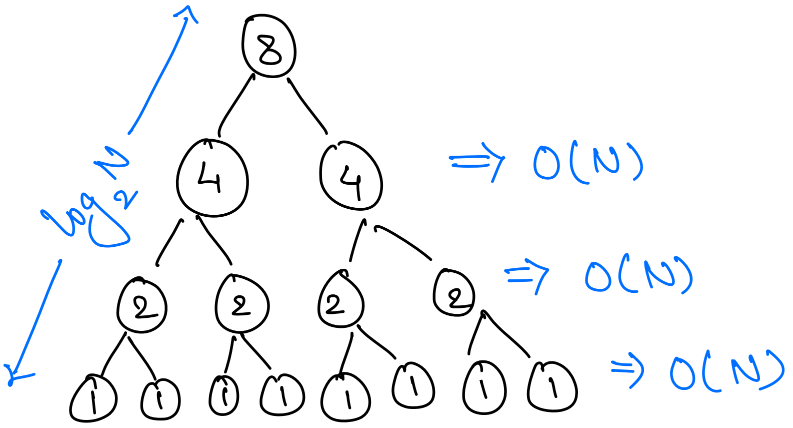
```
megeSort ( int A[] , int l, int r){
        if ( l== r)
              return;
        int  mid = (l+r)|2 ;
        mergesort (A , l, mid);
        mergeSort ( A, mid+1, r);
        merge ( A , l, mid , r)
}
```

Recurrence Relation :-

H/W $\quad T(N) = 2T(N|2) + N$

$$\boxed{TC : O(N \log N)}$$

Recursion Tree



TC: $O(N * \log_2 N)$

HW

SC : $O(N)$