Q.1 Given 2 Arrays A & B.
Count the no. of pairs i, j s.t A[i] > B[j].

$$A : \left\{ \begin{array}{ccc} 0 & 1 & 2 \\ 7 & 3 & 5 \end{array} \right\}$$

$$B : \left\{ \begin{array}{ccc} 0 & 1 & 2 \\ 2 & 0 & 6 \end{array} \right\}$$

$$\begin{array}{c} (i, j) \\ \downarrow \quad \downarrow \\ A[i] > B[j] \end{array}$$

(7,2) (7,0) (7,6)
(3,2) (3,0) (5,2)
(5,0)

} 7 pairs.

Quiz

$$A : \left[\begin{array}{ccc} 0 & 1 & 2 \\ 1 & 3 & 6 \end{array} \right]$$

$$B : \left[\begin{array}{ccc} 0 & 1 & 2 \\ 2 & 4 & 9 \end{array} \right]$$

(3,2) (6,2) (6,4) ⇒ 3 pairs.

Quiz

$$A : \left\{ \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 2 & 4 & 4 & 5 \end{array} \right\}$$

$$B : \left\{ \begin{array}{ccc} 0 & 1 & 2 \\ 3 & 2 & 9 \end{array} \right\}$$

(4,3) (4,2)
(4,3) (4,2)
(5,3) (5,2)

} 6 pairs.

Brute force :

⇒ for every element in A, check all the elements in B.

TC : O(N·M)

$A : \{ \overset{0}{7}\ \ \overset{1}{3}\ \ \overset{2}{5} \}$

$B : \{ \overset{0}{2}\ \ \overset{1}{0}\ \ \overset{2}{6} \}$

↓ sort

$A : \{ \overset{0}{3}\ \ \overset{1}{5}\ \ \underset{i}{\overset{2}{7}} \}$

$B : \{ \overset{0}{0}\ \ \overset{1}{2}\ \ \overset{2}{6} \}\ \ j$

ele	Pair
0	3
2	3
6	1
	7

A[i] > B[j] ⇒ index j will make a pair with all indices [i, N-1].

↳ $N-\cancel{1}-i+\cancel{1}$

⇒ $N-i$

Ex

$A : \{ \overset{0}{2}\ \ \overset{1}{4}\ \ \overset{2}{4}\ \ \overset{3}{5} \}\ \ i$

$B : \{ \overset{0}{2}\ \ \overset{1}{3}\ \ \underset{j}{\overset{2}{9}} \}$

ele	count
2	3
3	3
	6 pairs.

1) Sort(A) ⇒ $N \log N$
   Sort(B) ⇒ $M \log M$

2)
```
Count = 0
while( i< N && j< M){
    if (A[i] > B[j]){
        count += (N-i);
        j++
    }
    else {
        i++;
    }
}
```

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

$O(N\log N + M\log M)$

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

(Merge Sort)

Q. Given an Array of size N, count the no. of pairs i, j s.t $i < j$ && $A[i] > A[j]$.

Google
Amazon
Netflix
MS| Arcesium
Adobe...
...

Inversion Count

$$\begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \{10 & 3 & 8 & 15 & 6 & 12 & 2 & 18 & 7 & 1\} \end{array}$$

(10,3)
(10,8)
(10,6)
(10,2)
(10,7)
(10,1)

(3,2)
(3,1)

(8,6)
(8,2)
(8,7)
(8,1)

(15,6)
(15,12)
(15,2)
(15,7)
(15,1)

(6,2)
(6,1)

(12,2)
(12,7)
(12,1)

(2,1)

(18,7)
(18,1)

(7,1)

⇒ 26 Pairs.

Quiz

A : [8 4 2 1]

(8,4)
(8,2)
(8,1)

(4,2)
(4,1)

(2,1) ⇒ 6 pairs.

Quiz

[4 4 4 4 4]

$i < j$ && $A[i] > A[j]$

⇒ 0

Quiz A: $\begin{array}{ccc} 0 & 1 & 2 \\ 3 & 1 & 2 \end{array}$

$\left.\begin{array}{l}(3,1)\\(3,2)\end{array}\right\}$ 2

Brute force ⇒ $O(N^2)$

$i < j \;\&\&\; A[i] > A[j]$

$\{\begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 10 & 3 & 8 & 15 & 6 & 12 & 2 & 18 & 7 & 1 \end{array}\}$

Ⓐ $\{\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 10 & 3 & 8 & 15 & 6 \end{array}\}$

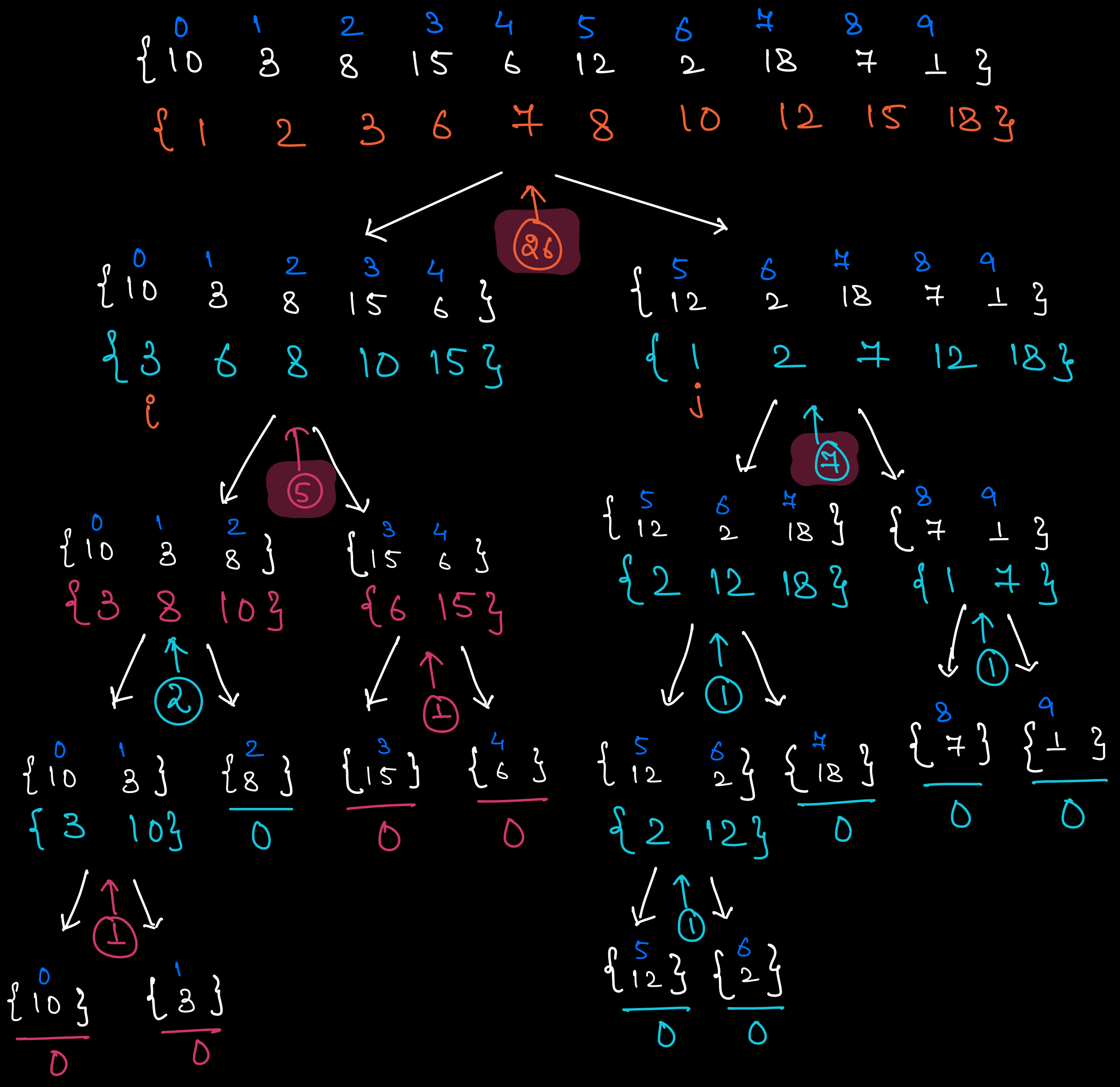
Ⓑ $\{\begin{array}{ccccc} 5 & 6 & 7 & 8 & 9 \\ 12 & 2 & 18 & 7 & 1 \end{array}\}$

Ⓐ $\{3 \quad 6 \quad 8 \quad 10 \quad 15\}$ i

Ⓑ $\{1 \quad 2 \quad 7 \quad 12 \quad 18\}$ j

ele	Count
1	5 ⇒ (3,1) (6,1) (8,1) (10,1) (15,1)
2	5
7	3
12	1
	14

$$\text{Total \# of inversion pairs} = \underbrace{\text{Pairs in } (A)}_{\text{first half}} + \underbrace{\text{Pairs in } (B)}_{\text{second half}} + \underbrace{\text{Pairs b/w A \& B}}$$



int mergeSort(A[], s, e){

}

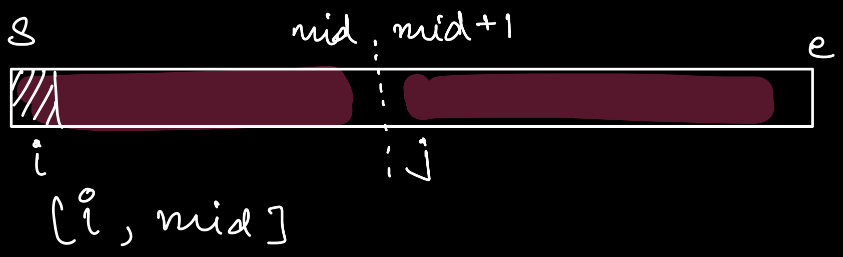
Assumption : mergeSort(A, s, e) fun^ will sort the array from s to e and returns the inversion count in A from s to e.

```
int count = 0;
int  mergeSort(A[], s, e){
     if(s == e){
          return 0;
     }
     mid = (s+e)|2
     count += mergesort(A, s, mid);  T(N/2)
     count += mergesort(A, mid+1, e); T(N/2)
     count += merge(A, s, mid, e);
                   ↳ O(N)
}
```

$T(N) = 2T(N/2) + O(N)$

→ TC : $O(N \log N)$

SC : $O(N)$



```
int   merge ( A[], s , mid , e ){
      // 1st half => s to mid
      // 2nd half => mid+1 to e.
      // Merge 2 halfs and count no. of inversions
      int C[e-s+1];
      i = s; j = mid+1;
      k=0;
      Count = 0;
      while( i <= mid && j <= e ){
              if (A[i] > A[j]){
                  Count += (mid- i+1);
                  C[k] = A[j]
                  j++, k++;
              }
              else{
                  C[k] = A[i]
                  k++, i++;
              }
      }
      while ( ______ ){
          [redacted]
      }
      while ( ______ ){
          [redacted]
      }
}
```

$$T(N) = 2T(N/2) + N$$

$$T(N) = 2\left[2T(N/4) + \frac{N}{2}\right] + N$$

$$= 4T(N/4) + 2N$$

$$= 4\left[2T(N/8) + \frac{N}{4}\right] + 2N$$

$$= 8T(N/8) + 3N$$

$$= 8\left[2T(N/16) + \frac{N}{8}\right] + 3 \cdot N$$

$$= 16T(N/16) + 4 \cdot N.$$

After k steps. :

$$\boxed{T(N) = 2^{k}\, T(N/2^{k}) + k \cdot N}$$

$$\frac{N}{2^{k}} = 1 \Rightarrow N = 2^{k}$$

$$\Downarrow$$

$$\log_2 N = \log_2 2^{k}$$

$$\log N = k$$

$$\boxed{\log_a a^{x} = x}$$

$$\boxed{2^{\log_2 x} = x}$$

$$T(N) = 2^{\log_2 N} \cdot \underbrace{T(1)}_{1} + \log N * N.$$

$$= N + N \log N$$

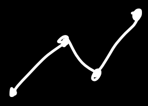
$$\Rightarrow \boxed{O(N \log N)}$$

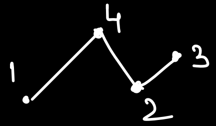
Q. Given an Array, convert into a wave array

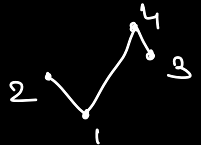
google.

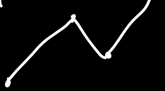
A: {1 2 3 4}

↓

{1 3 2 4} ✓ 

{1 4 2 3}  ✓

{2 1 4 3}  ✓

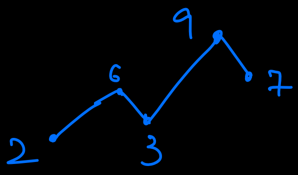
{1 3 2 4} 

⇒ lexicographically smallest array.

{1 3 2 4} < {1 4 2 3}

Ex

A : {3 9 7 6 2}

⇒ {2 6 3 9 7} 

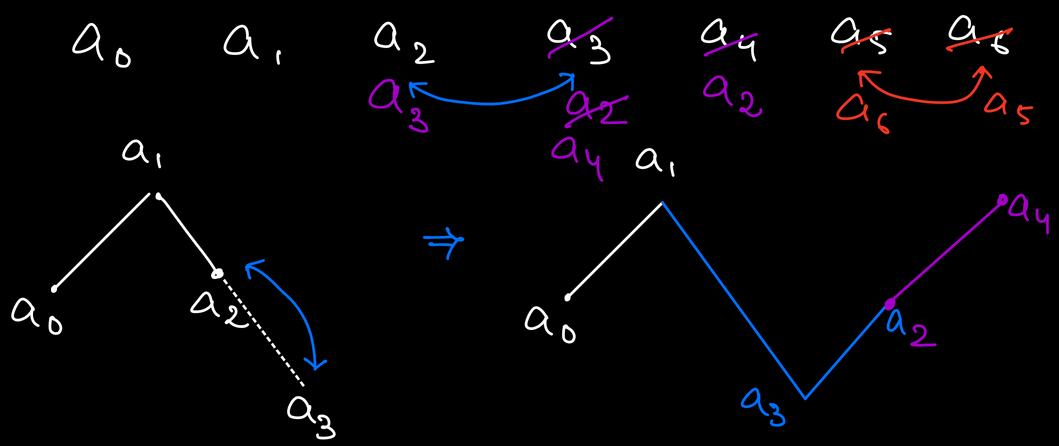
Idea

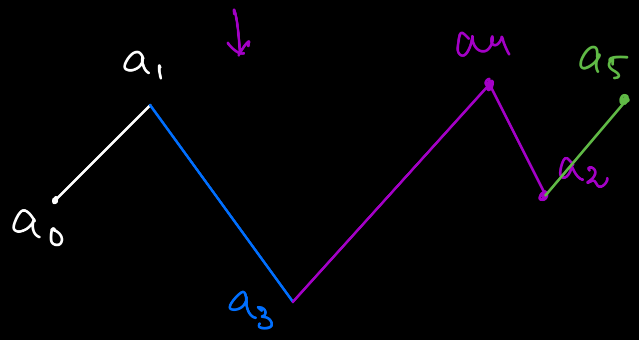
1) Sort

2) Swap(A[i], A[i+1]) from i=1

TC: O(NlogN)

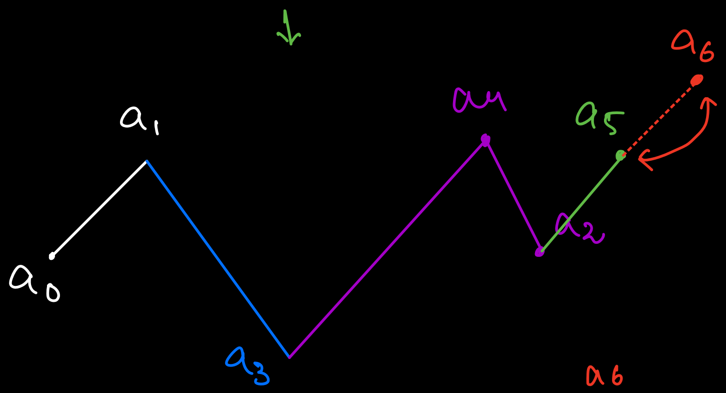
If smallest lenicographic ans is not required :



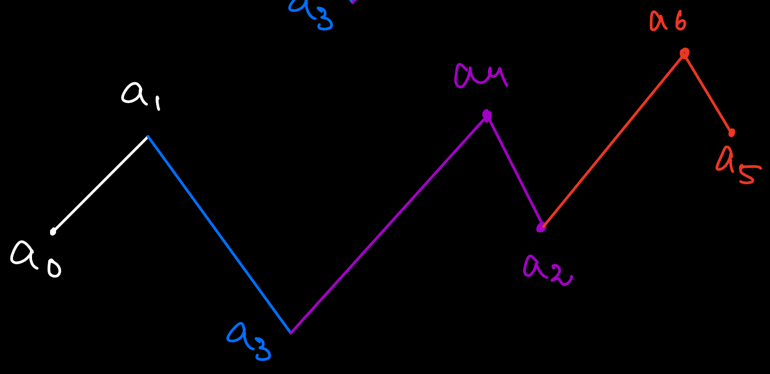


HW 30-40 mins.

Implement this approach.



$$\boxed{\begin{array}{l} TC: O(N) \\ SC: O(1) \end{array}}$$



——— * ———