



Q Given 2 arrays $A[N]$ & $B[M]$
Count the no of pairs i, j such that $A[i] > B[j]$

A : $\overset{0}{7}, \overset{1}{3}, \overset{2}{5}$
B : $2, \underset{\uparrow}{0}, 6$

$(7, 2) (7, 0) (7, 6)$

$(3, 2) (3, 0)$

$(5, 2) (5, 0)$

$\rightarrow 7$

Quiz

A : $\overset{0}{3}, \overset{1}{1}, \overset{2}{6}$
B : $2, 4, \underset{\uparrow}{9}$

$(3, 2)$ ✓

$(6, 2) (6, 4) \rightarrow 3$

Brute force TC : $O(NM)$

$\left\{ \begin{array}{l} A : \overset{\downarrow}{3}, \overset{\downarrow}{5}, \overset{\downarrow}{7} \\ B : \underset{\uparrow}{0}, 2, 6 \end{array} \right.$

$B[j]$

$0 \rightarrow 3$

$2 \rightarrow 3$

$6 \rightarrow 1$

A : 7, 8, 2, 4 \Rightarrow ⁰2, ¹4, ²7, ³8 ↓

B : 3, 5, 1, 10 \Rightarrow 1, ₃, 5, 10 ↑

$$[a, b] = b - a + 1$$

b	B[b]	Count	N=4
0	1	4	
1	3	3	
2	5	2	
3	10	0	
<hr/>		$\Sigma \rightarrow$	9

$$[a, (N-1)]$$

$$\Rightarrow \underline{N - a}$$

Step I

Sort A, Sort B
 $O(N \log N)$ $O(M \log M)$

Step II

• fin 2 pointers $a \rightarrow 0$, $b \rightarrow 0$

• if $(A[a] > B[b])$

$$\underline{\text{Count}} = \text{Count} + (N - a),$$

$$b++;$$

else

$a++;$

$O(M+N)$

$$TC : O(\underline{N \log N} + \underline{M \log M} + \underline{M+N}) \Rightarrow O(\underline{N \log N + M \log M})$$

$$SC : O(\underline{N + M})$$

$$O(\max(N, M))$$

(Assuming merge sort)

Google

Facebook

Amazon

Nelfin

Q Inversion Count

Given an array of N elements.

Count the no of pairs i, j
such that

$$i < j \\ \& \underline{A[i]} > \underline{A[j]}$$

A : ⁰10, ¹3, ²8, ³15, ⁴6, ⁵12, ⁶2, ⁷18, ⁸7, ⁹1

(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, 2) (15, 7) (15, 1) (15, 12) \Rightarrow 26

(6, 2) (6, 1)

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

(2, 1)

(18, 7) (18, 1)

(7, 1)

Quiz

A : ⁰3, ¹1, ²2

(3, 1) (3, 2) \rightarrow 2

Quiz

A : 8, 4, 2, 1

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

(4, 2) (4, 1)

(2, 1)

\rightarrow 6

Brute Force

TC : $O(N^2)$

A : ⁰10, ¹3, ²8, ³15, ⁴6

ⁱ
↓
A : 3, 6, 8, 10, 15

B : ⁰12, ¹2, ²18, ³7, ⁴1

^j
↓
B : 1, 2, 7, 12, 18

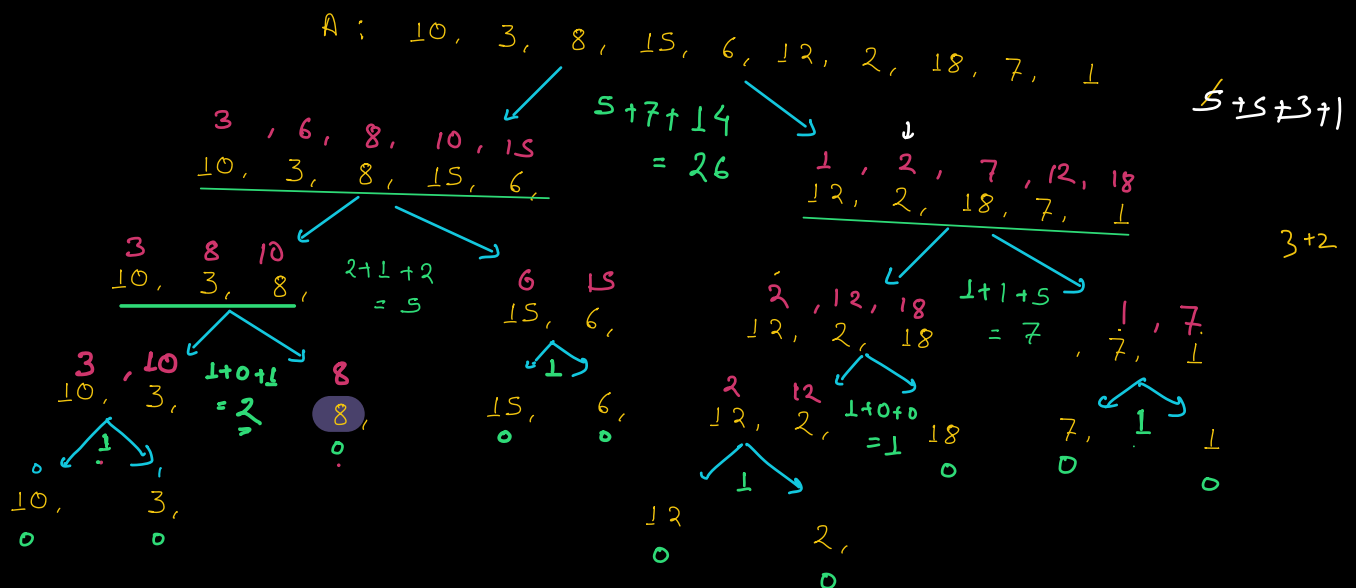
$A[i] > B[j]$

Total Pairs
(N)

= Pairs in array A
(N/2)

+ Pairs in array B
(N/2)

+ Pairs b/w A & B



```

int mergeSort (A[], s, e) {
    if (s == e) return 0;
    mid = (s+e)/2;
    l = mergeSort(A, s, mid);
    r = mergeSort(A, mid+1, e);
    return l+r+merge(A, s, mid, e);
}

```

```

int merge (A[], s, m, e) {
    T[e-s+1]; count = 0;
    i = 0; j = m+1, k = 0;
    while (i <= m && j <= e) {
        if (A[i] <= A[j]) { T[k] = A[i]; i++; k++; }
        else { T[k] = A[j]; j++; k++; }
        count += m-i+1;
    }
    while (i <= m) { T[k] = A[i]; i++; k++; }
    while (j <= e) { T[k] = A[j]; j++; k++; }
    for (i = 0; i <= (e-s); i++) {
        A[s+i] = T[i];
    }
    return count;
}

```

Google

Q Given an array.

Find the smallest sub-array after sorting which in asc order, the complete array will get sorted in asc order.

A : ⁰2, ¹6, ²4, ³8, ⁴10, ⁵9, ⁶15

s: 1, e: 5

Quiz

⁰1, ¹4, ²7, ³5, ⁴3, ⁵2, ⁶8, ⁷10, ⁸6, ⁹14, ¹⁰15

s: 1, e: 8

Quiz

⁰3, ¹4, ²1, ³2, ⁴6, ⁵7

s: 0, e: 3

TC: $O(N \log N)$ \Rightarrow Can we do better?

A : ⁰2, ¹6, ²4, ³8, ⁴10, ⁵9, ⁶15
↓ Sort
→ 2, 4, 6, 8, 9, 10, 15

TC: $O(N \log N)$

Step I : Copy array into a new array. Sort the new array

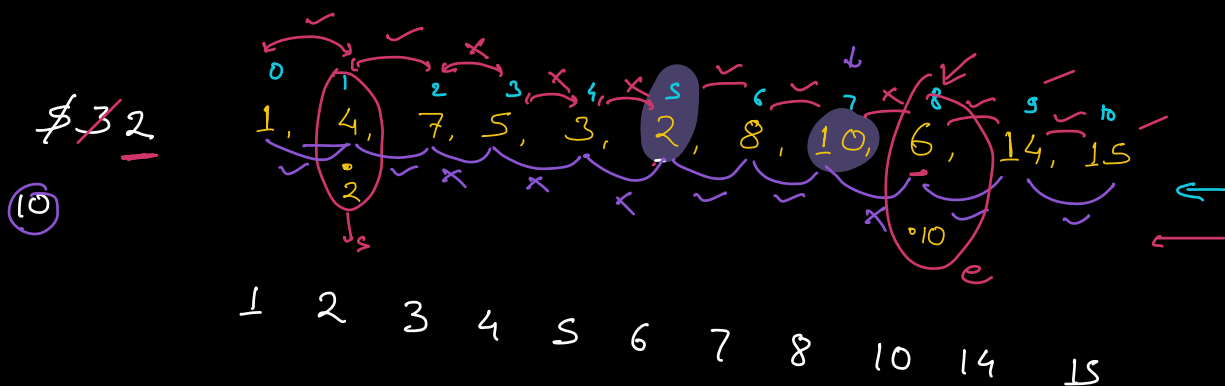
Step II :
s \rightarrow first mismatch from left
e \rightarrow first mismatch from right



$$\underline{A[i] < A[i+1]}$$



$$\underline{A[i] > A[i-1]}$$



s \rightarrow Index of smallest element
not present in its correct
position in sorted order.

e \rightarrow Index of largest element
not present in its correct
position in sorted order.

findUnsortedSubarray (A[]) {

// Find the smallest element not following $A[i] < A[i+1]$

min = ∞

i_max \rightarrow N-2

for (i=0; i < (N-1); i++) {

if ($A[i] > A[i+1]$) {

min = Math.min(min, $A[i+1]$);

}

// Find max value not following $A[i] > A[i-1]$

for (i=N-1; i > 0; i--) {

i_min = 1

if ($A[i] < A[i-1]$) {

max = Math.max(max, $A[i-1]$);

}

int s, e;

for (i=0; i < N; i++) {

if ($A[i] > \text{min}$) {

s = i;

break;

}

for (i=N-1; i >= 0; i--) {

if ($A[i] < \text{max}$) {

e = i;

break;

}

return s, e;

TC : $O(N)$

SC : $O(1)$

