Today's content

- Valid Pairs
- Inversion count
- Wave Array

Q) Given 2 arrays A[N] and B[M]

Count no. of pairs (i, j) such that A[i] > B[j]

Eg: A : [7 3 5] B : [2 0 6]
(0 1 2) (0 1 2)

<7,2> <3,2> <5,2>
<7,0> <3,0> <5,0>
<7,6>

{ ans → 7 }

Eg: A : [3 1 6] B : [2 4 9]
(0 1 2) (0 1 2)

<3,2> <6,2>
<6,4>

{ ans → 3 }

idea-1:

```
count : 0
for( i = 0 ; i < N ; i++) {
    for( j = 0 ; j < M ; j++) {
        if (A[i] > B[j]){
            count ++
        }
    }
}
return count
```

T.C → O(N*M)
S.C → O(1)

idea-2.

A[4]: [7 8 2 4]

A[4]: [2 4 7 8]

B[4]: [3 5 1 10]

B[4]: [1 3 5 10]

count: 4 +0 +3 +0 +2 +0 +0 +0

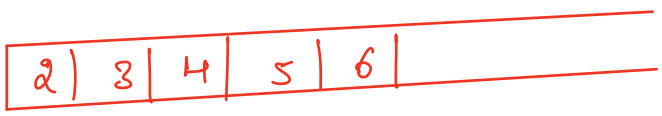
[ans = 9.]

A[4]: [8 5 3 10]

A[4]: [3 5 8 10]

B[4]: [6 4 2]

B[4]: [2 4 6]



count: 4 +0 +3 +0 +2

observation

B[j] < A[i].

→ Acquire element from B[].

→ Valid pairs with B[j] → A[i] to A[N-1].

[i, N-1] = N-1 - i + 1

= N - i

pseudo-code.

```
int validPairs ( A[], N , B[], M ){
    Sort (A[]);        ⟶  N log N
    Sort (B[]);        ⟶  M log M
    i = 0, j = 0, count = 0;
    while( i < N && j < M){
        if (A[i] <= B[j]){
            i++
        }
        else{
            count += N-i
            j++
        }
    }
    return count;
}
```

N+M.

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

T.C → $O(N \log N + M \log M + N + M)$

S.C → $O(N+M)$

a[] → [3 5 8 10]

b[] → [2 4 5 9]

| 2 | 3 | 4 | 5 | 5 |

count: 4 + 0 + 3 + 0 + 2

Google / Microsoft / De-Shaw / many more ... [Inversion Count].

Given A[N], find no. of pairs i, j such that i < j and A[i] > A[j]

Eg: arr[5] : [6 2 9 3 5]
0 1 2 3 4

[3] [0] [2] [0] [0]

{ans = 5}

Eg: arr[10] : [10 3 8 15 6 12 2 18 7 1]
0 1 2 3 4 5 6 7 8 9

[6] [2] [4] [5] [2] [3] [1] [2] [1] [0]

{ans = 26}

Idea:1. Consider all pairs

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

T.C → O(N²)
S.C → O(1)

arr[10]: [10 3 8 15 6 12 2 18 7 1]
0 1 2 3 4 5 6 7 8 9

A[] → [10 3 8 15 6]
0 1 2 3 4

B → [12 2 18 7 1]
5 6 7 8 9

⇒ [3 6 8 10 15]

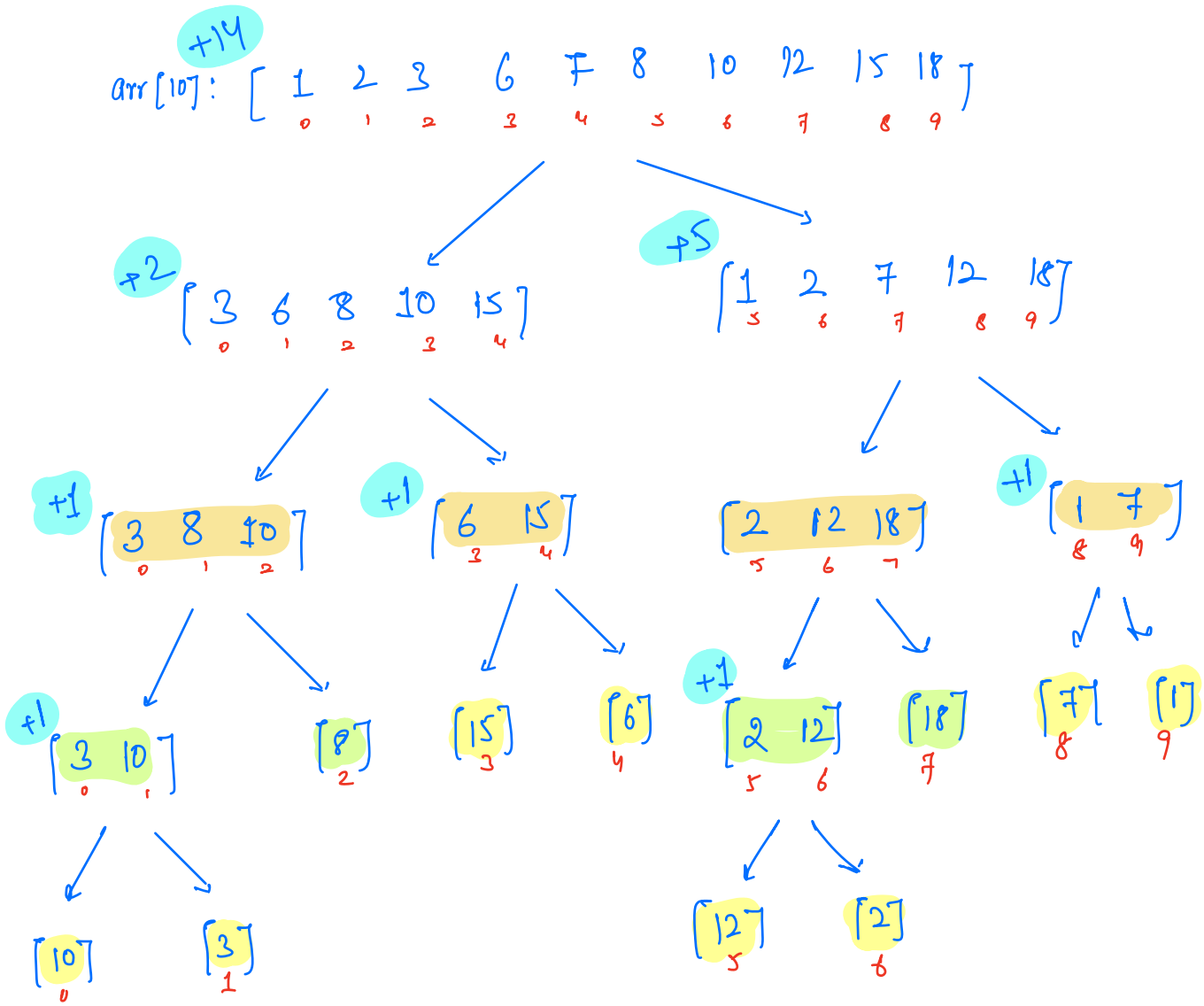
⇒ [1 2 7 12 18]

1	2	3	6	7	8	10	12	15	18

Count: +5 +5 +3 +1

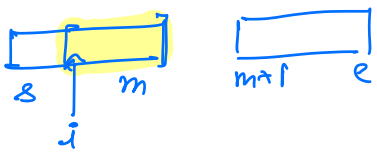
= 14

Total pairs = { Total pairs in A } + { Total pairs in B } +
{ Total pairs between A and B }




```
int inver ( arr, s, e) {
    if ( s == e) { return 0 }
    m = (s+e) / 2;
  left =  inver( arr, s, m);
  right = inver( arr, m+1, e);
  int c = merge (arr, s, m, e);
  return left + right + c
}
```

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



[q, b] → b - q + 1

(i, m] → m - i + 1

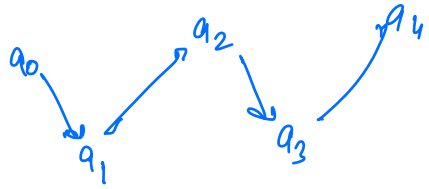
↳ only extra line.

T.C → O(N log N)

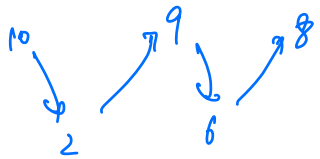
S.C → O(N)

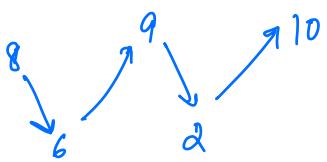
↳ while merging at every step, update our count.

Q) Given N distinct array elements. Re-arrange array in a wave form.



Eg → { 6 8 2 9 10 }

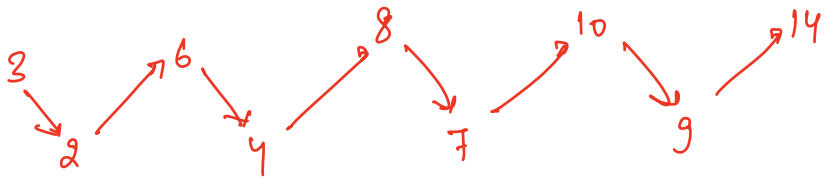




More than 1 wave is possible.
Q) Out of all these waves, return any wave.

Eg: [8 2 4 10 9 3 14 6 7]

arr → [2 3 4 6 7 8 9 10 14]



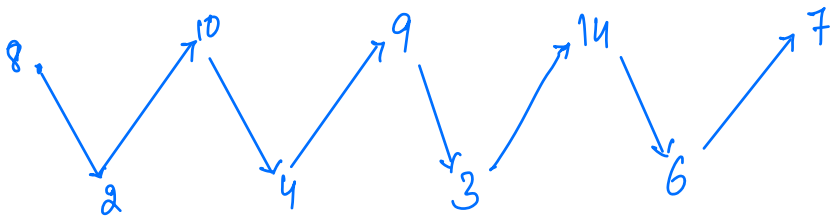
{lexicographically smallest wave}

idea →

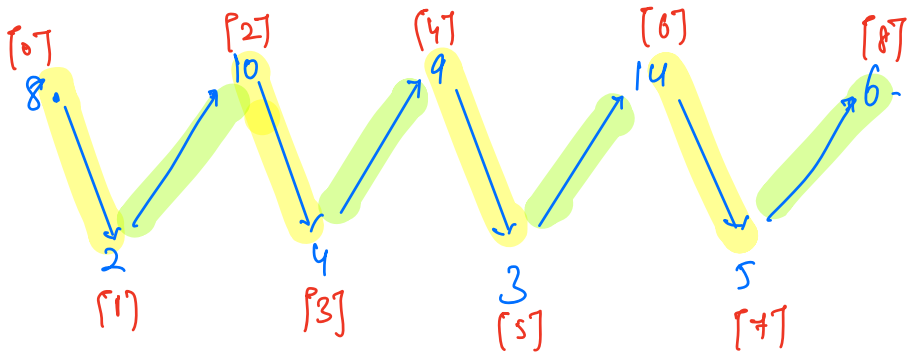
```
sort(arr);
for(i=0; i<N; i+=2){
    swap(arr, i, i+1);
}
```

{ T.C → O(N log N)
S.C → O(N) }

Eg: [8 2 ~~4~~ 10 ~~10~~ 4 9 3 ~~6~~ 14 ~~14~~ 6 7]



Eg: [8 2 ~~4~~ 10 ~~10~~ 4 9 3 ~~6~~ 14 ~~14~~ 6 ~~5~~ 5]



observation → if i is even, arr[i+1] should be smaller than arr[i]

if i is odd, arr[i+1] should be greater than arr[i]

→ otherwise, swap arr[i] with arr[i+1].

pseudo-code.

```
for ( i = 0 ; i < N-1 ; i++ ) {
    if ( i%2 == 0 ) {
        if ( arr[i+1] > arr[i] ) { swap (arr, i, i+1) }
    }
    else {
        if ( arr[i+1] < arr[i] ) { swap ( arr, i, i+1) }
    }
}
```

T.C → O(N)
S.C → O(1)

$n \to 12$, $m \to 30$.

$(n < m)$ swap $\Rightarrow$ $n = 30$, $m \to 12$.

$$
\begin{array}{r|l}
12 & 30 \\
 & 24 \\
\hline
 & 6 \\
\end{array}
\quad
\begin{array}{r|l}
 & 2 \\
\hline
6 & 12 \\
 & 12 \\
\hline
 & 0
\end{array}
$$

After sorting, consider any two consecutive nos.

$[25, 11, 3, 7, 44, 9]$ $\quad A[i] \,\%\, A[j]$

$\rightarrow [3, 7, 9, 11, 25, 44]$.