Merge Sort

Merge 2 Sorted Array

Suppose,

A=[1,2,3,0,0,0] — Extra-space given already

m=3 (Only 3 valid)

B=[2,5,6]

n=3

Merge 2 sorted form would be like

A=[1,2,3,2,5,6] but in sorted aswell

A= [1,2,2,3,5,6] — (Sorted)

A=[1,2,3]   B=[2,5,6]

C=[1,2,2,3,5,6]

We will see it in the decreasing/reverse order ⟵

idx ↑

| 1 | 2 | 3 | 0 | 0 | 0 | m

↓ i

idx = m+n-1

i=m-1, j=n-1

| 2 | 5 | 6 | n

↑ j

j < 0 ✗

→ j will be compared with i
greater will go to idx position (idx--, j--)

→ Then j will be compared with i
Then i with idx —

| 1 | 2 | 3 | 4 | 5 | 6 | m

i < 0

| 1 | 2 | 3 | n

Pseudocode :-

```
idx = m+n-1 , i = m-1 , j = n-1
while( i>=0 && j>=0){
    if(A[i] >= A[j]){
    A[idx] = A[i] }
        idx--
        i--
    else
    A[idx] = B[j]
        idx--
        j--
while (j >= 0){
    A[idx] = B[j];
    idx--
    j--
```

T.C
$O(m+n)$

Additional loops
if A/i gets empty but j/B
remain full

Next Permutation

A = [1,2,3] Return lexicographically next.

Next permutation must be in-place of existing array.

1,2,3	3,1,2
1,3,2	3,2,1
2,1,3	
2,3,1	

√ = greater ther

6 Possible Permutations

→ Brute Force Approach

① finding all the permutations

② find lex next permutation of our array.

→ Optimal Approach O(1) → T.C

O(n) → S.C

Common Observation Pivot = A[i] < A[i+1]

1,2,3,5,4   1 2 3 6 5 4 already largest   2 5 4 3

In decreasing order 1 2 4 6 5 3 to be smallest

1 2 4 3 5   1 2 4 3 5 6 have to reverse   1 3 2 4 5

Pivot is the decreased element if we start to come backward.

① Find the pivot element.

② Find the right most element > pivot

③ swap(RME, Pivot)

④ Reverse (pivot+1) to n-1

1 2 5 4 3 largest
P

1 3 5 4 2 ← have to make smallest
P reverse them

1 3 2 4 5 (Answer)

Pseudocode

5,4,3,2,1
decreasing
piv = -1
Reverse it

```
piv = -1
for(i=n-2 ; i>=0 ; i--){
    if (A[i] <A[i+1]){
        piv = i
        break;
    }
}
if (piv == -1)
    reverse array
    return
```

Edge case
reverse(A.begin(), A.end())

```
for(i=n-1, i>pivot; i--){
    A[i] > A[pivot]
        swap(A[i], A[pivot])
            break;
    }}
i = piv+1
j = n-1
while(i<=j){
    swap(A[i], A[j])
    i++
    j--
}
```