Today's content

- → Basics of Sorting
- → Selection sort
- → Merge two sorted arrays -->IMP
- → Merge Sort
- → Inversion count (Google, Amazon, G.S)

Sorting : Arranging data in a specific order based on parameter.

1, 2, 3, 5, 7, 10 → increasing (parameter → value)

9, 6, 3, -1, -5 → decreasing (parameter → value)

1, 7, 2, 9, 24 → increasing (parameter → count of factors)

factors → 1 2 2 3 8

Why sorting ? → searching of an element becomes easy & fast.

stable / unstable sorting → Two data points have same parameter value and the order is preserved before and after the sorting then this is called stable sorting.

Name.	Marks
Ashok	92
Shubham	45
Divya	100
Brian	45
Vitul	42

$\xrightarrow[\text{basis of marks}]{\text{sort on the}}$

Name	Marks.
Vitul	42
Shubham	45
Brian	45
Ashok	92
Divya	100

stable.

arr[] → [1 5 2 6 2]

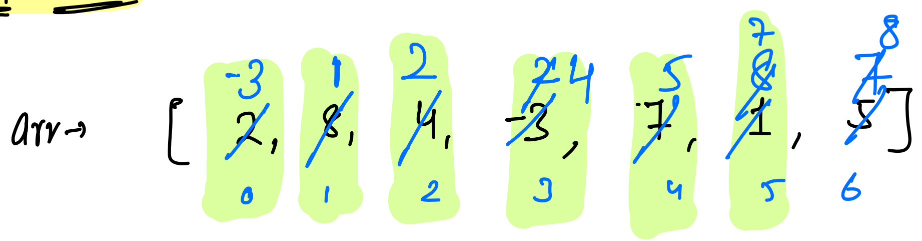
sort1. → 1 2 2 5 6 → stable sorting

sort2. → 1 2 2 5 6 → unstable sorting

In-place Sorting → Sorting the data without taking extra space.

S.C → O(1)

Selection sort



i	min	idx.	swap
1	1	5	swap (arr [i] with arr(idx])
2	2	3	swap(arr[2] with arr[3])
3	4	3	swap(arr[3] with arr[3])
4	5	6	swap (arr(4] with arr[6])
5	7	6	swap (arr [5] with arr[6])

code.

```
void selectionSort( int[] arr , int N){

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

        min = arr[i], idx = i

        for(j = i+1; j < N; j++){

            if (arr[j] < min ){
                min = arr[j];
                idx = j;
            }
        }

        swap (arr[i] with arr[idx]);
    }
}
```

min element

[i+1, N-1]

T.C → $O(N^2)$

S.C → $O(1)$

⇓

in-place ✓

Stable / unstable ?

2 4 2 1

⇓

1 4 2 2

⇓

1 2 4 2

⇓

1 2 2 4

⇒ unstable sorting

Q: Given two sorted arrays. Merge them & get a final sorted array.

$A[\,] \rightarrow [\underset{0}{2}\quad \underset{1}{3}\quad \underset{2}{7}\quad \underset{3}{12}\quad \underset{4}{20}\quad \underset{5}{24}\quad \underset{6}{29}] \Rightarrow N$

$B[\,] \rightarrow [\underset{0}{6}\quad \underset{1}{9}\quad \underset{2}{12}\quad \underset{3}{14}\quad \underset{4}{15}\quad \underset{5}{19}] \Rightarrow M.$

idea → Copy all the elements from A[] and B[] in C[] and sort C[].

T.C → $O((N+M)\log(N+M))$

S.C → $O(1)$

$i \downarrow$

idea → $A[\,] \rightarrow [\underset{0}{\cancel{2}}\quad \underset{1}{\cancel{3}}\quad \underset{2}{\cancel{7}}\quad \underset{3}{\cancel{12}}\quad \underset{4}{\cancel{20}}\quad \underset{5}{\cancel{24}}\quad \underset{6}{\cancel{29}}] \Rightarrow N$

$B[\,] \rightarrow [\underset{0}{\cancel{6}}\quad \underset{1}{\cancel{9}}\quad \underset{2}{\cancel{12}}\quad \underset{3}{\cancel{14}}\quad \underset{4}{\cancel{15}}\quad \underset{5}{\cancel{19}}] \Rightarrow M.$

$j \uparrow$

$C[N+M]: [\underset{0}{2}\quad \underset{1}{3}\quad \underset{2}{6}\quad \underset{3}{7}\quad \underset{4}{9}\quad \underset{5}{12}\quad \underset{6}{12}\quad \underset{7}{14}\quad \underset{8}{15}\quad \underset{9}{19}\quad \underset{10}{20}\quad \underset{11}{24}\quad \underset{12}{29}]$

$\uparrow k$

#code

```
int[]  mergedTwoSortedArrays ( int[] A, int[] B, int N, int m){

        int C[N+M];
        i=0 , j=0, K=0;
        while( i<N && j<m){
                if( A[i] ≤ B[j]){
                        C[K]= A[i]
                        i++ ; K++
                }
                else{
                        C[K]= B[j];
                        j++ , K++ ;
                }
        }

        while( i<N){
                C[K]= A[i]
                i++, K++
        }
        while( j<m){
                C[K]= B[j];
                j++, K++
        }

        return C[];
}
```

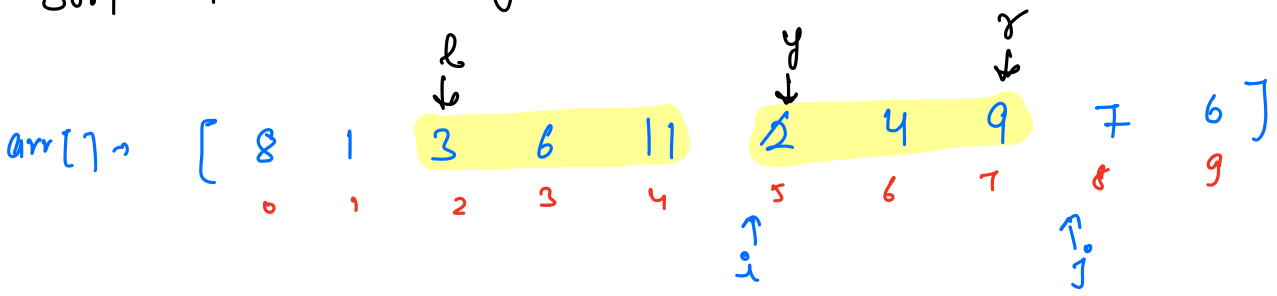
T.C → O(N+m)

S.C → O(1)

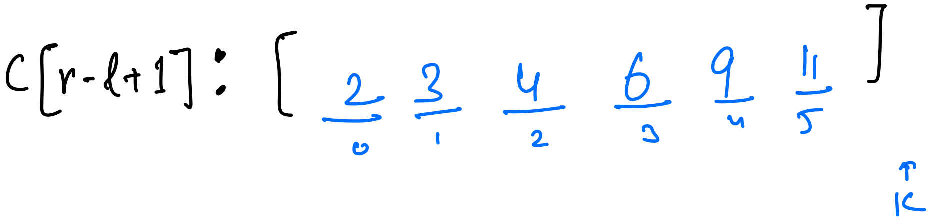
→ Single array and 3 integers → l, y, r. [l<y<r]

{ subarray from l to y-1
subarray from y to r } ⇒ sorted.

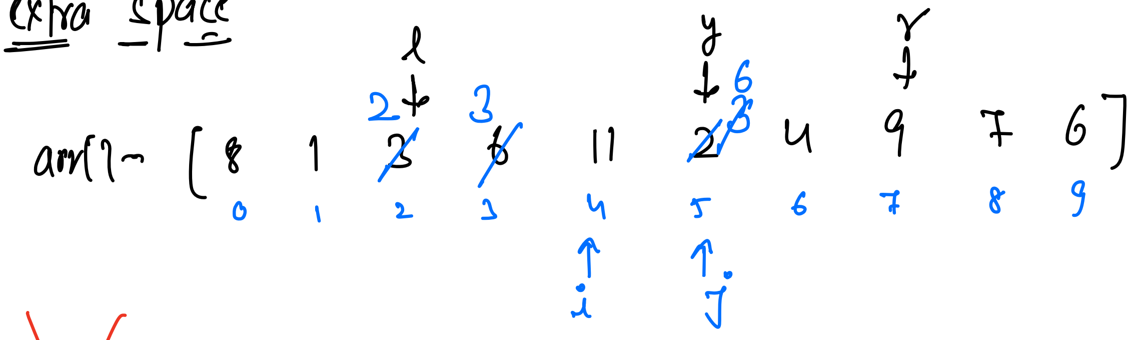
Sort the subarray from l to r.



l = 2
y = 5
r = 7



Without extra space



A[j] < A[i] ⇒ swap A[i] with A[j]

i++.

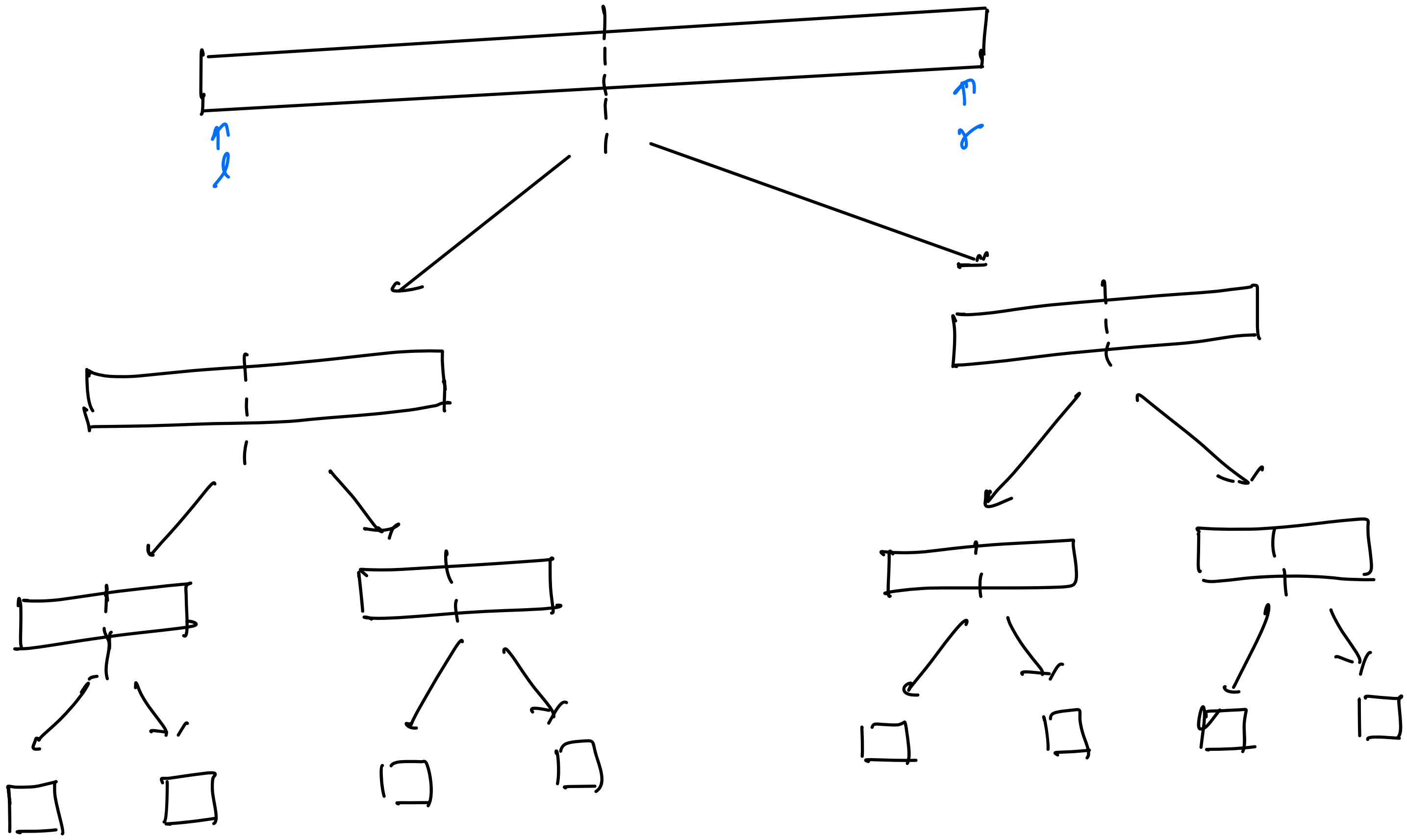
code →

```
int[]  mergedTwoSortedsubArrays( int [] A , int l, int y, int r){

        int C[r-l+1];
        i= l, j= y, K=0;
        while( i < y  && j ≤ r ){

              if( A[i]  ≤ A[j]){
              {
                     C[K] = A[i]
                      i++ ; K++
              }
              else{
              {
                     C[K] = A[j];
                      j++ , K++ ;
              }
        }

        while( i < y ){
        {
              C[K]= A[i]
               i++, K++
        }
        while( j ≤ r){
        {
              C[K] = A [j];
               j++, K++
        }
        //copy the elements from C[] to A[]

         K=0
         for( i = l;  i  ≤ r;  i++){
         {
                A[i] = C[K]
                K++
         }
}
```

$$T.C \to O(r-l+1) \to O(N)$$

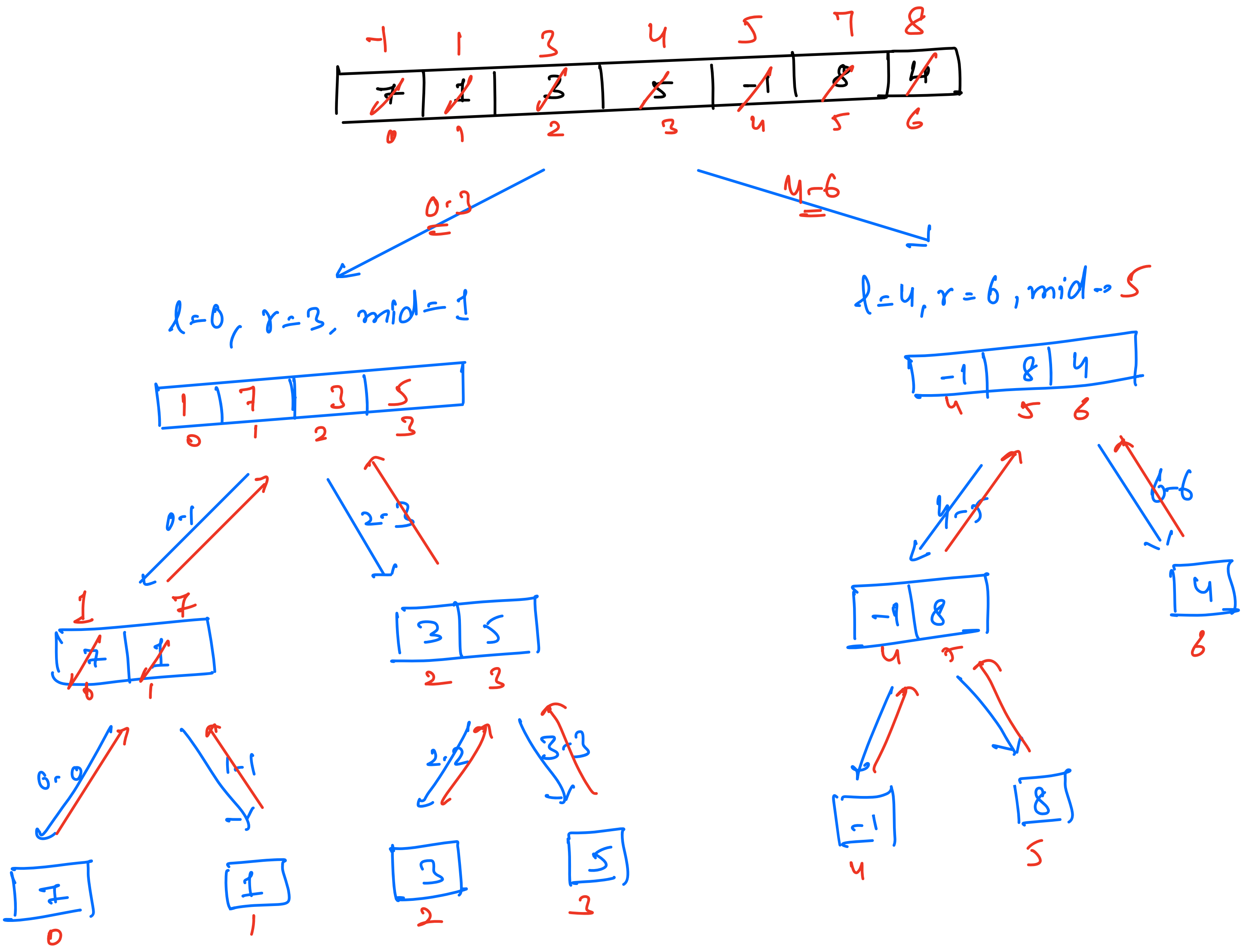
$$S.C \to O(r-l+1) \to O(N)$$

Merge Sort (Divide and Conquer)



code :-

```
void    mergeSort( arr, l, r) {
{
        if( l == r) { return }
        mid = (l+r)/2;
        mergeSort( arr, l, mid);
        merge Sort (arr, mid+1, r);
        mergeTwoSortedSubarrays( arr, l ,mid+1 , r);
}
```



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

$$\left[T(N/2) = 2T(N/4) + N/2\right] * 2$$

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

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

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

$$4T(N/4) = 8T(N/8) + N$$

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

$$T(N) = 16\,T(N/16) + 4N$$

$$T(N) = 32\,T(N/32) + 5N$$

$$\vdots$$

$$T(N) = 2^k\,T\left(\frac{N}{2^k}\right) + k \cdot N \qquad \underline{T(1) = 1.}$$

$$\left\{ \frac{N}{2^k} = 1 \Rightarrow N = 2^k \Rightarrow \log_2 N = k \right\}$$

$$T(N) = N \cdot \cancelto{1}{T(1)} + \log_2 N \cdot N$$

$$= N + N \log_2 N$$

$$\boxed{\begin{array}{l} T.C \rightarrow O(N \log N) \\ S.C \rightarrow O(N) \end{array}}$$

Inversion Count → Google, Amazon, G.S.

Given an array of size N. Find the no. of pairs i, j such that $i < j$ and $A[i] > A[j]$

↳ inversion pairs.

arr →

[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	= 26.

T.C → $O(N^2)$ S.C → $O(1)$

idea →

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

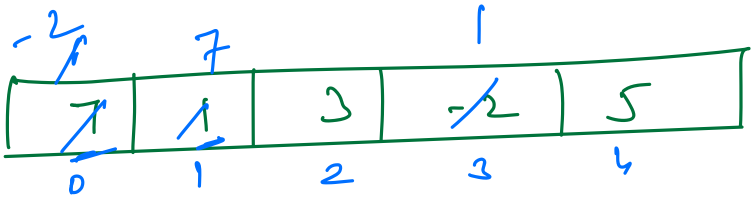
$3+2+3+4+2$

total inversion pairs = I.P in first half + I.P in second half + I.P across the subarrays.

$= 5 + 7 + 14 = 26.$

[idea → Merge Sort:]

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


-2
1
7
1
7
1
3
-2
5
0
1
2
3
4