

Today's content

- Merge 2 sorted arrays
- Merge 2 sorted subarrays
- Merge Sort
- Comparator.

Q Given 2 sorted arrays $A[N]$, $B[M]$ merge, create & return a new sorted array.

idea-1.

$A[N]$, $B[M]$, $C[N+M]$

$a[3] : \{-1, 4, 8\}$

$b[2] : \{2, 9\}$

$c[] : \{-1, 2, 4, 8, 9\}$

N : ① Copy all elements from $A[]$ to $C[]$

M : ② Copy all elements from $B[]$ to $C[]$

$(N+M) \log_2(N+M)$: ③ Sort $C[]$

T.C $\rightarrow O((N+M) \log_2(N+M))$
S.C $\rightarrow O(1)$

$a1[7] : \{ \cancel{5}_0, \cancel{1}_1, \cancel{2}_2, \cancel{4}_3, 9_4, 10_5, 15_6 \}$

\uparrow
 i

$a2[5] : \{ \cancel{-3}_0, \cancel{-2}_1, \cancel{-1}_2, \cancel{0}_3, \cancel{1}_4, ? \}$

\uparrow
 j

$C : \begin{bmatrix} -5 & -3 & -2 & -1 & -1 & 2 & 4 & 6 & 7 & 9 & 10 & 15 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{bmatrix}$

$a1[i]$ and $a2[j]$

pseudo-code

```
int[] mergeSortedArrays ( int[] a1, N, int[] a2, int M) {
```

```
    int i = 0, j = 0, k = 0
```

```
    int[] c = new int [N+M];
```

```
    while ( i < N && j < M) {
```

```
        if (a1[i] <= a2[j]) {
```

```
            c[k] = a1[i]
```

```
            i++, k++
```

```
        } else {
```

```
            c[k] = a2[j]
```

```
            j++, k++
```

```
        }
```

```
    }
```

```
    while ( i < N) {
```

```
        c[k] = a1[i]
```

```
        i++, k++
```

```
    }
```

```
    while ( j < M) {
```

```
        c[k] = a2[j]
```

```
        j++, k++
```

```
    }
```

```
    return c;
```

```
}
```

$N=5, M=3$

a1 →

-2	1	5	9	14
0	1	2	3	4

 ↑
i

a2 →

-5	1	3
0	1	2

 ↑
j

c →

-5	-2	1	1	3	5	9	14
0	1	2	3	4	5	6	7

 ↑
k

i = 2 ~~3~~ 5

j = 3

k = 8 ~~7~~ 8

T.C → $O(N+M)$

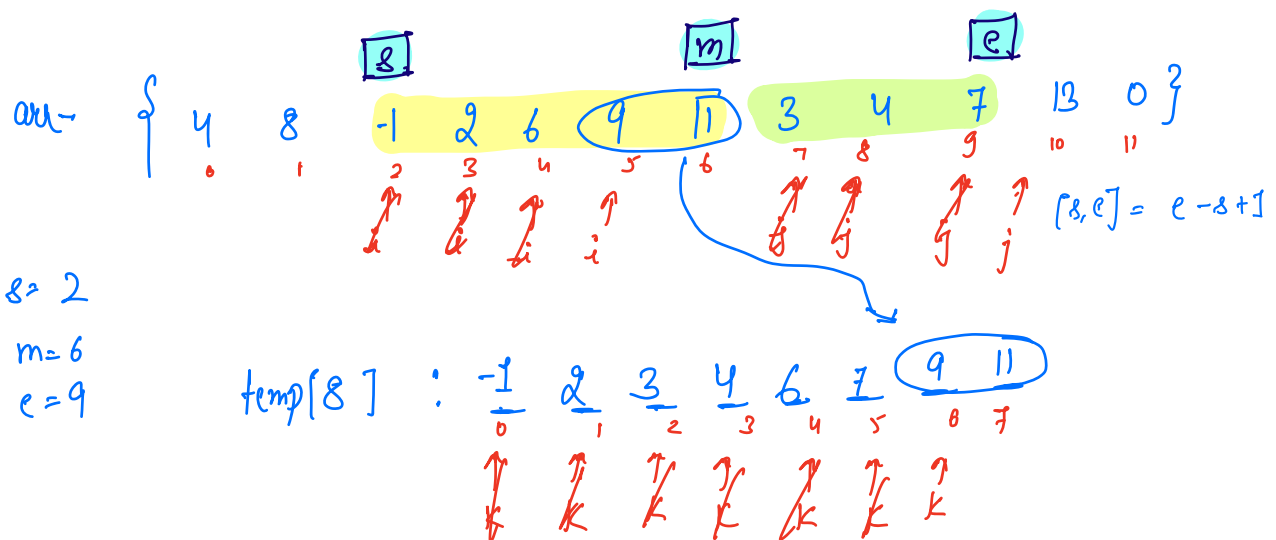
S.C → $O(1)$

Q: Given N array elements. and 3 input indices s, m, e

Also, given subarray $[s, m]$ is sorted $s < m < e$

subarray $[m+1, e]$ is sorted

→ Sort the subarray from s to e .



{ $i = s$, $j = m+1$, $k = 0$ }

→ compare arr[i] with arr[j]

arr → { 4 8 -1 2 3 4 6 7 9 11 13 0 }

pseudo-code

```
void merge ( arr, s, m, e ) {  
    i = s, j = m+1, k = 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[k] = arr[j]  
            j++, k++  
        }  
    }  
    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]  
    }  
}
```

$i \rightarrow s+1$

$arr[s] = temp[0]$


$arr[s+1] = temp[1]$


$arr[s+2] = temp[2]$

$T.C \rightarrow O(N)$
 $S.C \rightarrow O(N)$ \rightarrow { Not possible }

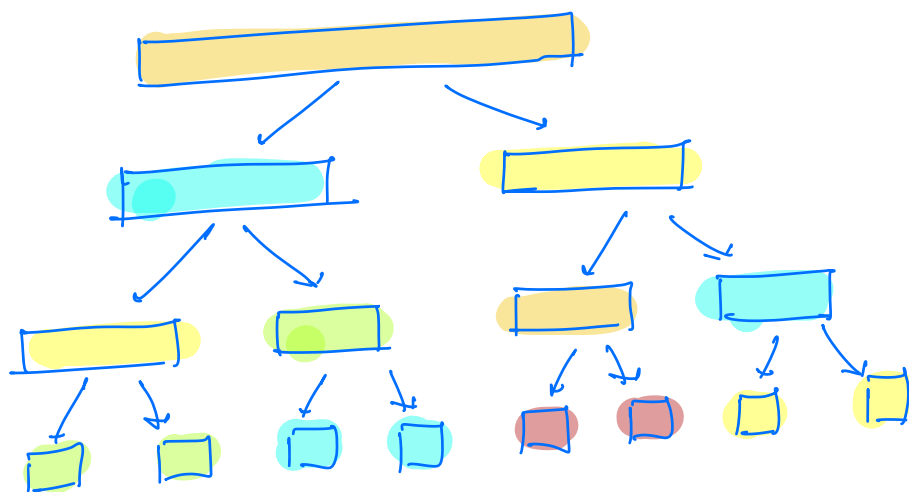
Q. Given N elements array. Sort it. BS / S.S / I.S
 $N = 100$

Case-1.  $\therefore N^2 : 10^4$

Case-2.  $\left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + N$
 $= \frac{N^2}{4} + \frac{N^2}{4} + N$
 $= \frac{N^2}{2} + N$
 $= 5100$

Case-3.  $\left(\frac{N}{4}\right)^2 * 4 + \frac{N}{2} + \frac{N}{2} + N$
 $= \frac{N^2}{4} + 2N$
 $= 2700$

The moment when size of subarray = 1, we can't divide further.

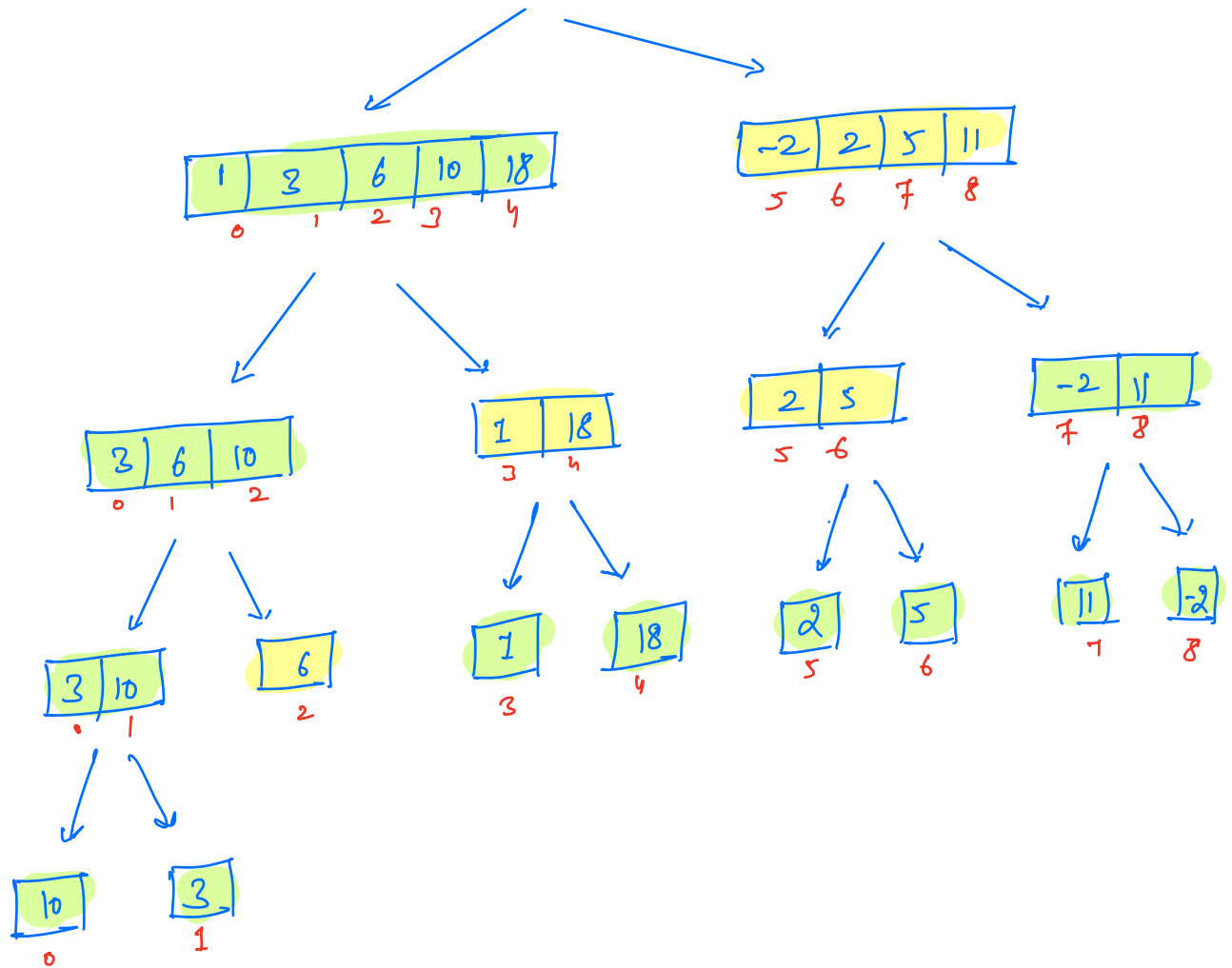


idea of Merge Sort.

arr[9]:

-2	1	2	3	5	6	10	11	18
0	1	2	3	4	5	6	7	8

[0-8].



// Ass^m → Given subarray from s to e, it should sort subarray from s to e.

```
void MergeSort ( arr, s, e ) {  
    if ( s == e ) { return }  
    m = (s+e) / 2 ;  
    MergeSort ( arr, s, m );  
    MergeSort ( arr, m+1, e );  
    merge ( arr, s, m, e );  
}
```

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

$$\{ T(N) = 2T(N/2) + N \}$$

$$T(1) = 1.$$

```
void merge ( arr, s, m, e ) {
```

```
    i = s, j = m+1, k = 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[k] = arr[j]
```

```
            j++, k++
```

```
        }
```

```
    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]
```

```
    }
```

```
}
```

Comparator → [Revise].

↳ using it, you can get your own desired sorted order.

Time Complexity.

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

$$\uparrow T(N/2) = 2T(N/4) + \frac{N}{2}$$

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

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

$$\uparrow T(N/4) = 2T(N/8) + \frac{N}{4}$$

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

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

Generalisation

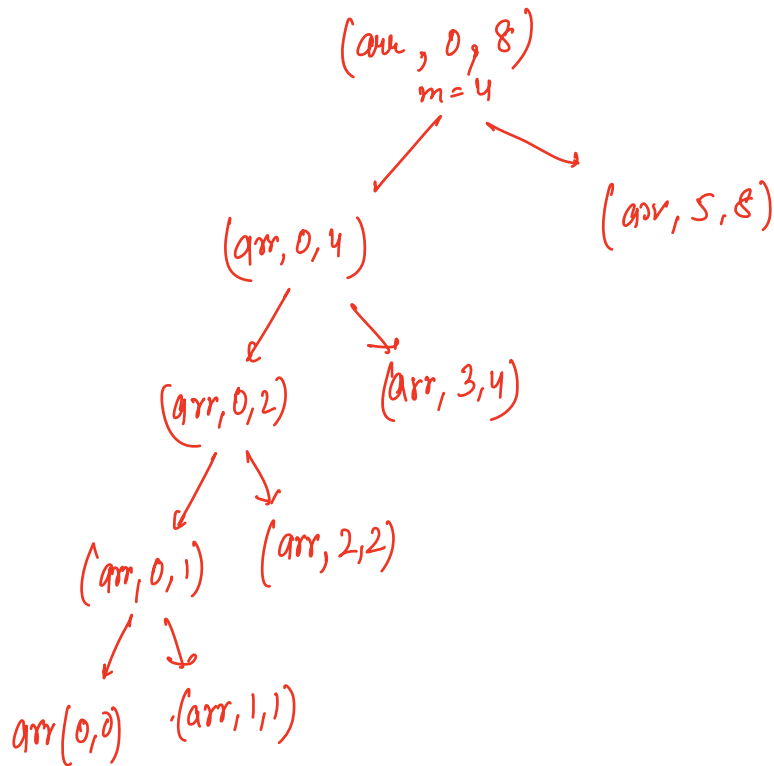
$$T(N) = 2^K T\left(\frac{N}{2^K}\right) + K \cdot N$$

$$\therefore \frac{N}{2^K} = 1 \Rightarrow [N = 2^K] \Rightarrow [K = \log_2 N]$$

$$T(N) = N T(1) + \log_2 N \cdot N$$

$$T(N) = N \log_2 N + N$$

$$\begin{aligned} T.C &\rightarrow O(N \log_2 N) \\ S.C &\rightarrow O(\log N + N) \\ &= O(N) \end{aligned}$$



n
 \downarrow
 $n/2$
 \downarrow
 $n/4$
 \vdots
 1

[\rightarrow Sorting 2 : All problems.
 \rightarrow Sorting 3 : Algorithms.]

B.S.

\rightarrow bubbling process
 \rightarrow largest bubble will burst first
 \downarrow
 " element " go to its
 \downarrow
 correct posⁿ first.

I.S.

\rightarrow Arrange cards.

Selection

\rightarrow select the min & send it to its correct position

Merge Sort

\rightarrow merging at every step.

9:05. — ^{break} 5-6 → 11:30

A, B.

$$\left[\begin{array}{l} A \% x == 0 \quad \{x \text{ is a factor of } A\} \\ x \text{ and } B \text{ are co-prime. } \gcd(x, B) = 1 \end{array} \right]$$

x = max value

A = 30 → 1, 2, 3, 5, 6, 10, 15, 30.

B = 12 → 1, 2, 3, 4, 6, 12.

[x = 5]

$\gcd(A, B) \rightarrow \gcd(30, 12) \rightarrow \underline{6}$.

~~36~~ (5)

$$\left\{ \begin{array}{l} \gcd(A, B) = 6. \\ \gcd(\underbrace{\text{factor of } A}_x, B) = 1. \end{array} \right\}$$

$\{A = A/6.\}$

$$\left\{ \begin{array}{l} \text{while}(\gcd(A, B) \neq 1) \{ \\ \quad \text{val} = \gcd(A, B) \\ \quad A = A / \gcd(A, B) \\ \} \end{array} \right.$$