

Merge Sort

16/Apr/2023

Divide & Conquer: \downarrow

<https://upload.wikimedia.org/wikipedia/commons/c/cc/Merge-sort-example-300px.gif>

Q.) Merge two sorted array into a 3rd array in sorted order.

eg: A

1	3	5	7	9
---	---	---	---	---

 sorted

B

2	4	12
---	---	----

 sorted

} I/p

C

1	2	3	4	5	7	9	12
---	---	---	---	---	---	---	----

 } O/p

A

0	1	2	3	4
1	3	5	7	

$\text{len}(A) = 4$

B

\cancel{j}	\cancel{j}	\cancel{j}	\cancel{j}	i		
2	4	6	8	12	16	17

$\text{len}(B) = 7$

$\text{len}(C) = 4 + 7 = 11$

C

0	1	2	3	4	5	6	7	8	9	10	11
\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}	\cancel{k}
1	2	3	4	5	6	7	8	12	16	17	

```

static int[] mergeSortedArrays(int arr1[], int arr2[])
{
    int merged[] = new int [arr1.length + arr2.length];
    int i, j, k;
    i = j = k = 0;

```

```

    while( i < arr1.length && j < arr2.length)
    {
        if (arr1[i] < arr2[j])
        {
            merged[k] = arr1[i];
            i++;
            k++;
        }
        else
        {
            merged[k] = arr2[j];
            j++;
            k++;
        }
    }

```

```

    while( i < arr1.length)
    {
        merged[k] = arr1[i];
        k++; i++;
    }

```

```

    while( j < arr2.length)
    {
        merged[k] = arr2[j];
        k++;
        j++;
    }

```

3

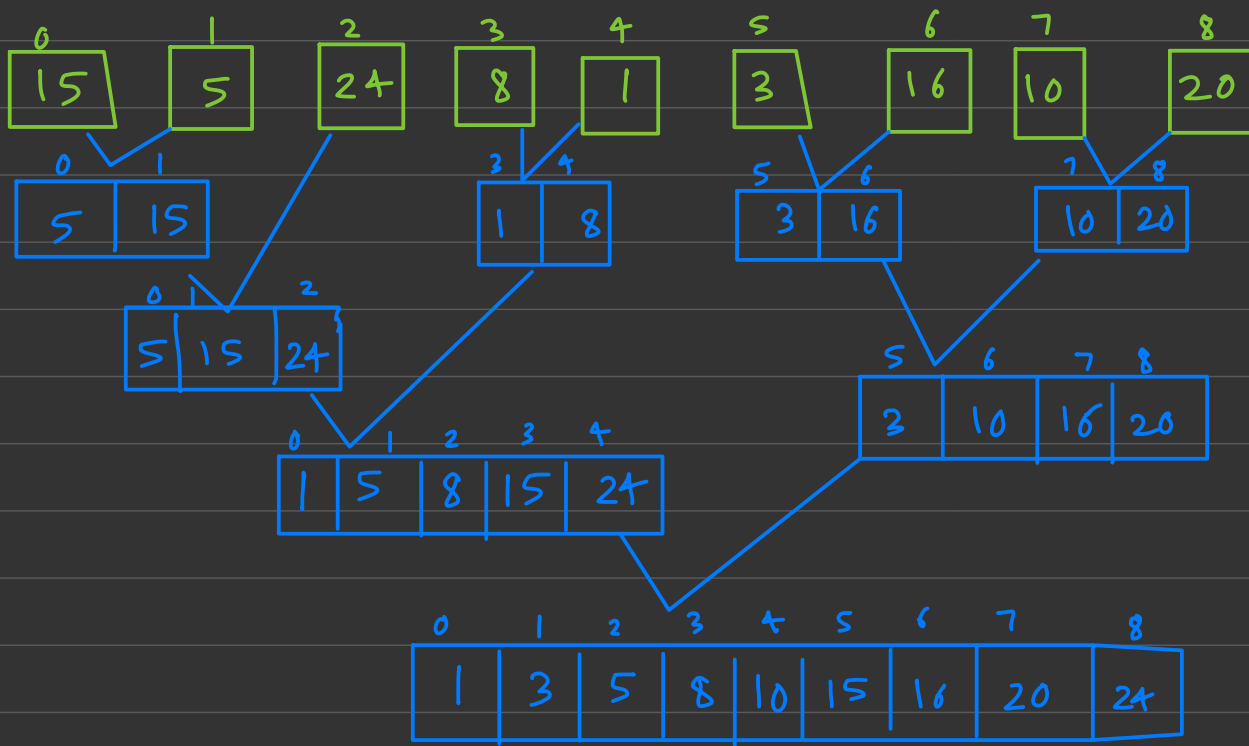
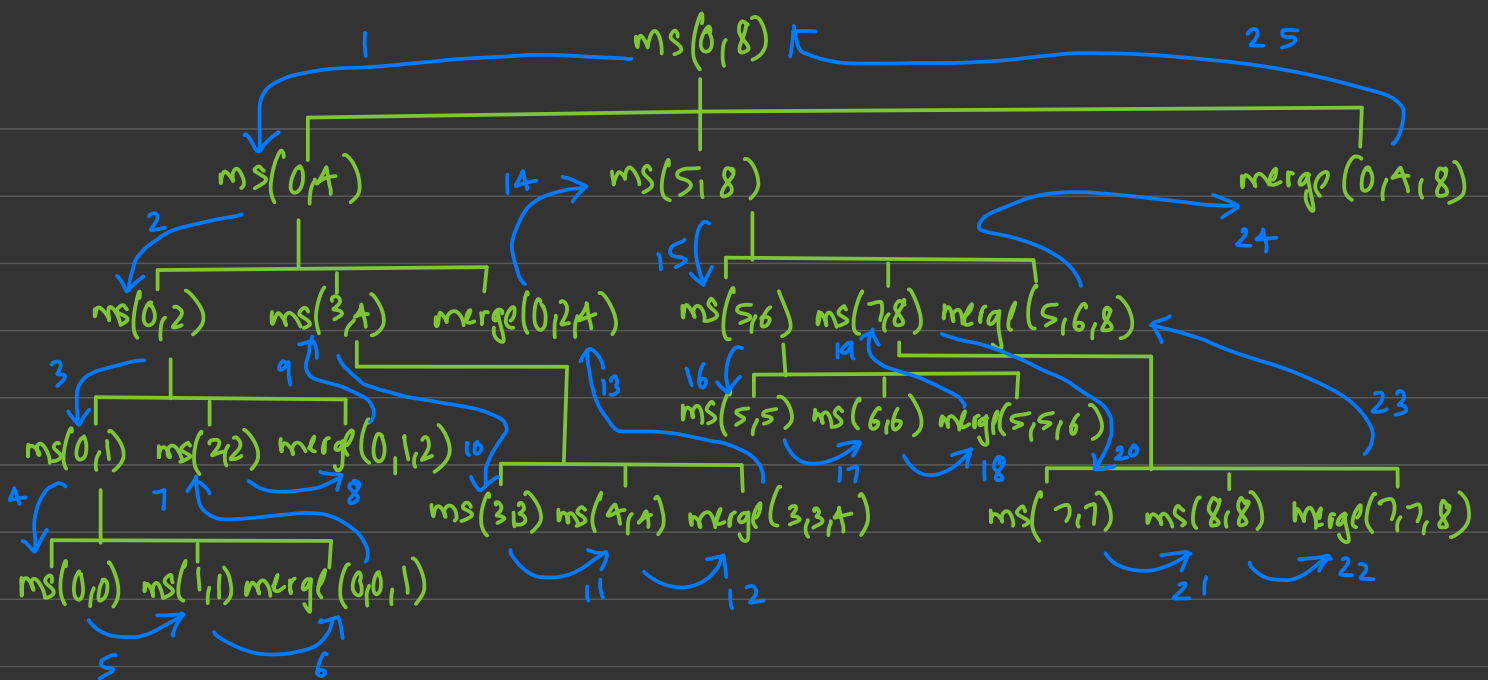
The diagram illustrates the merging process. At the top, two arrays are shown: `arr1` with elements `1, 3, 5` and `arr2` with elements `2, 4, 7, 9`. Above `arr1`, indices `i` and `j` are marked above the first and second elements respectively. Above `arr2`, indices `i` and `j` are marked above the first and second elements respectively. Below these, a third array `mergedarr` is shown with elements `1, 2, 3, 4, 5, 7, 9`. Above `mergedarr`, indices `k` are marked above each element. To the right of the arrays, the lengths are noted: `len(arr1) = 3`, `len(arr2) = 4`, and `len(mergedarr) = 7`. A small inset box in the top right corner contains the following text:

```

// arr1 = [1, 3, 5]
// arr2 = [2, 4, 7, 9]
// mergedarr = [1, 2, 3, 4, 5, 7, 9]

```

To be continued . . .



————— X —————