Merge two Sorted Arrays

Given two sorted arrays, the task is to merge them in a sorted manner.
Examples:

> ***Input***: *arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}*
> ***Output***: *arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}*
>
> ***Input***: *arr1[] = { 5, 8, 9}, arr2[] = {4, 7, 8}*
> ***Output***: *arr3[] = {4, 5, 7, 8, 8, 9}*

Method 1: (O(n1 * n2) Time and O(n1+n2) Extra Space)

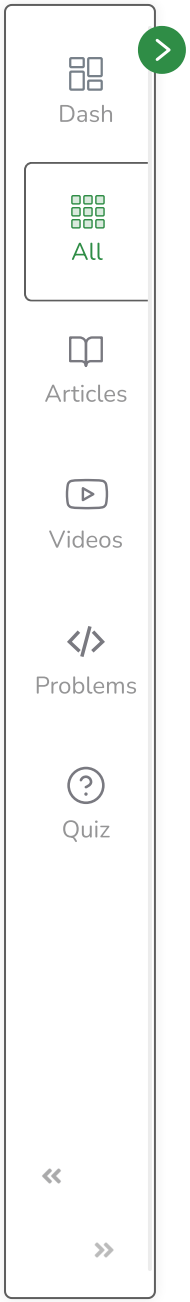
1. Create an array arr3[] of size n1 + n2.
2. Copy all n1 elements of arr1[] to arr3[]
3. Traverse arr2[] and one by one insert elements (like insertion sort) of arr3[] to arr1[]. This step take O(n1 * n2) time.

We have discussed implementation of above method in Merge two sorted arrays with O(1) extra space
Method 2: (O(n1 + n2) Time and O(n1 + n2) Extra Space)
The idea is to use Merge function of Merge sort.

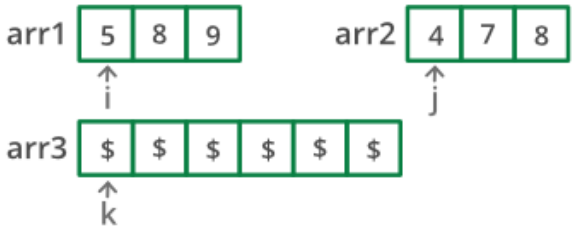
1. Create an array arr3[] of size n1 + n2.
2. Simultaneously traverse arr1[] and arr2[].
    - Pick smaller of current elements in arr1[] and arr2[], copy this smaller element to next position in arr3[] and move ahead in arr3[] and the array whose element is picked.

3. If there are remaining elements in arr1[] or arr2[], copy them also in arr3[].

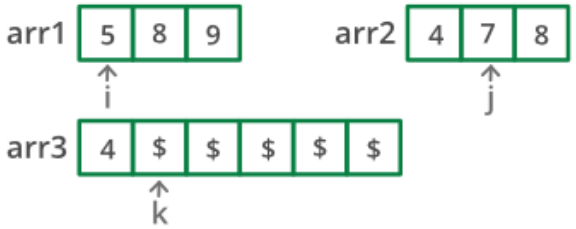
Below image is a dry run of the above approach:


Dash
All
Articles
Videos
Problems
Quiz

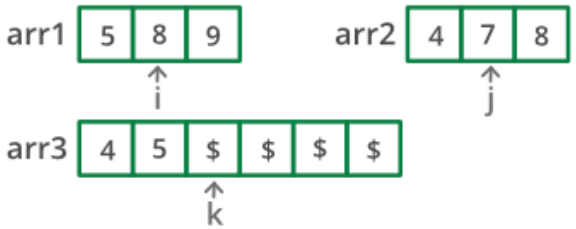
Initially:

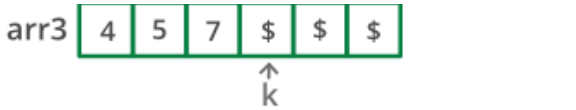


Step 1:

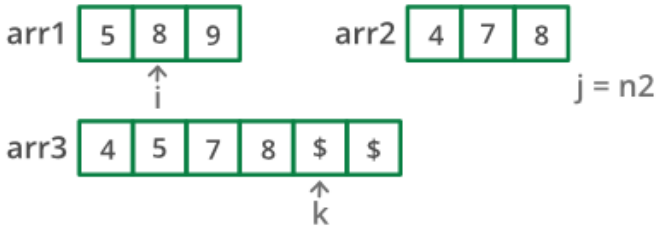


Step 2:



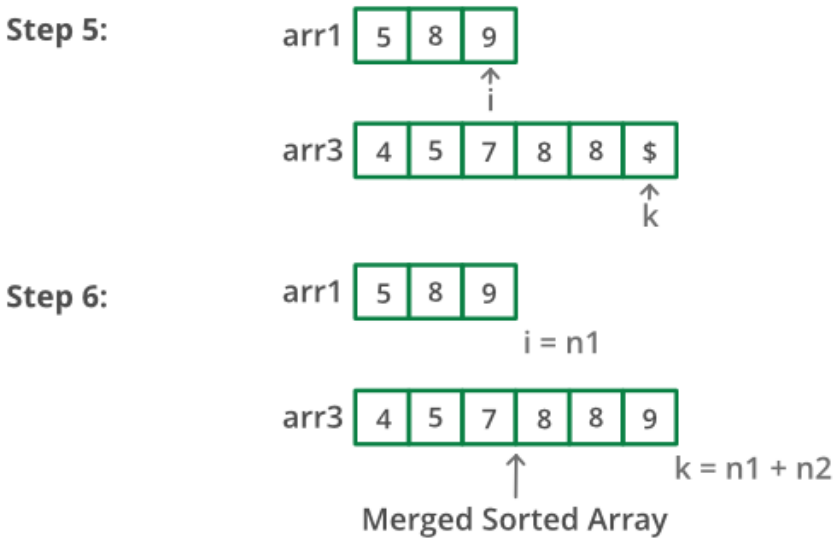


Step 4:



The first while loop breaks

Second while loop copies all elements from arr1 to arr3



Below is the implementation of the above approach:

C++

```
#include <bits/stdc++.h>
using namespace std;

void mergeArrays(int arr1[], int arr2[], int n1, int n2, int arr3[]) {
    int i = 0, j = 0, k = 0;

    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j])
            arr3[k++] = arr1[i++];
```

```
        else
            arr3[k++] = arr2[j++];
    }

    while (i < n1)
        arr3[k++] = arr1[i++];

    while (j < n2)
        arr3[k++] = arr2[j++];
}

int main() {
    int arr1[] = {1, 3, 5, 7};
    int n1 = sizeof(arr1) / sizeof(arr1[0]);

    int arr2[] = {2, 4, 6, 8};
    int n2 = sizeof(arr2) / sizeof(arr2[0]);

    int arr3[n1+n2];
    mergeArrays(arr1, arr2, n1, n2, arr3);

    cout<<"Array after merging"<<"\n";
    for (int i = 0; i < n1+n2; i++)
        cout<<arr3[i]<<" ";

    return 0;
}
```

Output

```
Array after merging
1 2 3 4 5 6 7 8
```