

# Solved Problem - Merge Sorted Arrays

In this lesson, we'll discuss how to merge two sorted arrays.

## We'll cover the following

- Problem statement
- Sample
- Solution
- Time complexity

## Problem statement #

Given two sorted arrays,  $A[]$  and  $B[]$ , of sizes  $N$  and  $M$  respectively, merge them into a single array of size  $N + M$  and print the array.

### Input format

The first line consists of two space-separated integers  $N, M$  ( $1 \leq N, M \leq 10^5$ ).

The second line consists of  $N$  space-separated integers representing the array  $A[]$  ( $1 \leq A[i] \leq 10^5$ ).

The third line consists of  $M$  space-separated integers representing the array  $B[]$  ( $1 \leq B[i] \leq 10^5$ ).

### Output format

Print a single line of output containing the  $N + M$  integers representing the merged and sorted array  $C[]$ .

## Sample #

Input:

```
4 4
2 3 4 6
```

2 3 4 6  
1 5 7 8

Output

1 2 3 4 5 6 7 8

Solution #

We use 2 pointers  $i$  and  $j$  for  $A[]$  and  $B[]$  respectively.

At each step, we copy the smaller of  $A[i], B[j]$  to the current end of  $C[]$ . Keep doing this until we reach the end of A or B. After that only one array remains (either  $A$  or  $B$ ), copy all the remaining elements to C.

See the illustration for better understanding.

2

3

4

6

1

5

7

8

1

i=0 j=0 { A[0] > B[0] => C[0] = B[0] }

1 of 8

2

3

4

6

1

5

7

8

1

2

i=0 j=1 { A[0] < B[1] => C[1] = A[0] }

2 of 8

2

3

4

6

1

5

7

8

1

2

3

1	2	3					
---	---	---	--	--	--	--	--

$i=1 \ j=1 \ \{ A[1] < B[1] \Rightarrow C[2] = A[1] \}$

3 of 8

2	3	4	6
---	---	---	---

1	5	7	8
---	---	---	---

1	2	3	4				
---	---	---	---	--	--	--	--

$i=2 \ j=1 \ \{ A[2] < B[1] \Rightarrow C[3] = A[2] \}$

4 of 8

2	3	4	6
---	---	---	---

1	5	7	8
---	---	---	---

1	2	3	4	5			
---	---	---	---	---	--	--	--

$i=3 \ j=1 \ \{ A[3] > B[1] \Rightarrow C[4] = B[1] \}$

5 of 8

2	3	4	6
---	---	---	---

1	5	7	8
---	---	---	---

1	2	3	4	5	6		
---	---	---	---	---	---	--	--

$i=3 \ j=2 \ \{ A[3] < B[2] \Rightarrow C[5] = A[3] \}$

6 of 8

2	3	4	6
---	---	---	---

1	5	7	8
---	---	---	---

1	2	3	4	5	6	7	
---	---	---	---	---	---	---	--

$j=2 \ \{ C[6] = B[2] \}$

7 of 8

2	3	4	6
---	---	---	---

1	5	7	8
---	---	---	---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

$j=3 \{ C[7] = B[3] \}$

8 of 8

—

[ ]

main.cpp

input.txt

All code files are copied to end of the page...



## Time complexity #

Since we iterate both the arrays exactly once, the time complexity of the solution is  $O(N + M)$ .

In the next lesson, we'll discuss a very useful prefix-sum technique.

## Code Files Content !!!

↓  
 ↓ ↓

```
-----
|  main.cpp [1]
|  -----
#include
```

```

#include
using namespace std;

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

    // one of A or B will exhaust first, we'll copy the remaining to C
    while (i < N) // B exhausted
        C[m++] = A[i++];
    while (j < M) // A exhausted
        C[m++] = B[j++];
}

int main() {
    ifstream cin("input.txt");

    int N, M;
    cin >> N >> M;
    int A[N], B[M], C[N + M];
    for (int i = 0; i < N; i++) cin >> A[i];
    for (int i = 0; i < M; i++) cin >> B[i];

    merge(A, B, C, N, M);

    for (int i = 0; i < N + M; i++)
        cout << C[i] <<" ";

    return 0;
}

```

```

-----
|  input.txt [1]
-----

```

```

4 4
2 3 4 6
1 5 7 8

```

```

*****

```