

Merge sorted array

inp1: 1, 2, 3
inp2: 2 5 6

length of inp1: $m+n$
length of inp2: n

idea:

When two array is sorted, their last element is greater so, we compare last element of both array and greater will assign last position of First array. and so on.

Algorithm: we use pointer to solve this problem in $O(m+n)$ time complexity and $O(1)$ space complexity.



Pseudo code:

$i = m - 1, j = n - 1, k = m + n - 1$

while $i \geq 0$ and $j \geq 0$ and $i < j$

while $i \geq 0$ and $j \geq 0$

if $\text{in}[\text{p1}[i]] > \text{in}[\text{p2}[j]]$

$\text{in}[\text{p1}[k--]] = \text{in}[\text{p1}[i--]]$

else

$\text{in}[\text{p1}[k--]] = \text{in}[\text{p2}[j--]]$

while $j \geq 0$

$\text{in}[\text{p1}[k--]] = \text{in}[\text{p2}[j--]]$



Dry Run

inpt 1	inpt 2	i	j	k	condn	Action
1, 2, 3	2, 5, 6	2	2	5	$3 > 6$	6 at 5
1, 2, 3, 0, 0, 5	2, 5, 6	0	1	4	$3 > 2$	3 at 4
1, 2, 3, 0, 3, 6	2, 5, 6	1	1	3	$2 > 2$	2 at 3
1, 2, 3, 2, 3, 6	2, 5, 6	1	0	2		

inpt 1: [1|2,3|0|0|0]

inpt 1	inpt 2	i	j	k	condn	Action
1, 2, 3	2, 5, 6	2	2	5	$3 > 6$	6 at
1, 2, 3	2, 5, 6	2	1	4	$3 > 5$	5 at
1, 2, 3	2, 5, 6	2	0	3	$3 > 2$	3 at
1, 2, 3	2, 5, 6	1	0	2	$2 > 2$	2 at

final output [1|2|3|3|5|6.]