Merge Sorted Array

Given two sorted integer arrays *nums1* and *nums2*, merge *nums2* into *nums1* as one sorted array.

Note:
You may assume that *nums1* has enough space (size that is greater or equal to *m* + *n*) to hold additional elements from *nums2*. The number of elements initialized in *nums1* and *nums2* are *m* and *n* respectively.

Solution 1

```
class Solution {
public:
    void merge(int A[], int m, int B[], int n) {
        int i=m-1;
        int j=n-1;
        int k = m+n-1;
        while(i >=0 && j>=0)
        {
            if(A[i] > B[j])
                A[k--] = A[i--];
            else
                A[k--] = B[j--];
        }
        while(j>=0)
            A[k--] = B[j--];
    }
};
```

written by leetchunhui original link here

Solution 2

This code relies on the simple observation that once all of the numbers from `nums2` have been merged into `nums1`, the rest of the numbers in `nums1` that were not moved are already in the correct place.

```
class Solution {
public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
        int i = m - 1, j = n - 1, tar = m + n - 1;
        while (j >= 0) {
            nums1[tar--] = i >= 0 && nums1[i] > nums2[j] ? nums1[i--] : nums2[j--
];
        }
    }
};
```

written by deck original link here

Solution 3

```java
public void merge(int A[], int m, int B[], int n) {
    int i=m-1, j=n-1, k=m+n-1;
    while (i>-1 && j>-1) A[k--]= (A[i]>B[j]) ? A[i--] : B[j--];
    while (j>-1)         A[k--]=B[j--];
}
```

written by annafan original link here