Merge Sorted Array

Question: Given two sorted integer arrays A and B, merge B into A as one sorted array.

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

Solutions:

```
class Solution(object):
    def merge(self, A, m, B, n):
        """
        :type A: List[int]
        :type m: int
        :type B: List[int]
        :type n: int
        :rtype: void Do not return anything, modify A in-place instead.
        """
        indexA = m-1;
        indexB = n-1;
        while indexA >=0 and indexB>=0:
            if A[indexA] > B[indexB]:
                A[indexA+indexB+1] = A[indexA]
                indexA -= 1
            else:
                A[indexA+indexB+1] = B[indexB]
                indexB -= 1
        while indexB >= 0:
            A[indexB] = B[indexB]
```

```
        indexB -= 1
```

```
Solution().merge([1],1,[],0)
```

All you need is understanding the algorithm, don't try other cases.

In case you're wondering, if the requirements do not include modifying an array in-place. This is how you merge arrays:

```
arr1 = [1,2,3,10,20]
arr2 = [2,2,10,60]
result = sorted(arr1 + arr2)
print ( result )
```