Question:

**88. 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:

public class Solution {

public void merge(int[] nums1, int m, int[] nums2, int n) {

// base/corner cases

if (nums1 == null || nums2 == null) {

return;

}

// preprocessing

int i = shiftToEnd(nums1, m);

// main case

int j = 0;

for (int k = 0; k < nums1.length; k++) {

// base/corner cases

if (i >= nums1.length) {

nums1[k] = nums2[j];

j++;

}

else if (j >= n) {

nums1[k] = nums1[i];

i++;

}

else { // main case

int elem1 = nums1[i];

int elem2 = nums2[j];

if (elem1 <= elem2) {

nums1[k] = elem1;

i++;

} else {

nums1[k] = elem2;

j++;

}

}

}

}

private int shiftToEnd(int[] nums1, int m) {

int endIndex = nums1.length - 1;

for (int i = m-1; i >= 0; i--) {

nums1[endIndex] = nums1[i];

endIndex--;

}

return endIndex+1;

}

}

Sample Run:

[ 3 4 7 \_ \_ \_ \_ ]

[ 1 2 5 6 ]

| |

[ \_ \_ \_ \_ 3 4 7 ]

[ 1 2 5 6 ]

|

| |

[ 1 \_ \_ \_ 3 4 7 ]

[ 1 2 5 6 ]

|

| |

[ 1 2 \_ \_ 3 4 7 ]

[ 1 2 5 6 ]

|

| |

[ 1 2 3 \_ 3 4 7 ]

[ 1 2 5 6 ]

|

| |

[ 1 2 3 4 3 4 7 ]

[ 1 2 5 6 ]

|

| |

[ 1 2 3 4 5 4 7 ]

[ 1 2 5 6 ]

|

||

[ 1 2 3 4 5 6 7 ]

[ 1 2 5 6 ]

|

||

[ 1 2 3 4 5 6 7 ]

[ 1 2 5 6 ]

|