CS 245 - Lab 3 Fall, 2021

Lab 3 - Merge Arrays

The goal of this assignment is to demonstrate your mastery of sorting. Download your code here.

Background

- You are given two arrays `arr1` and `arr2`, both arrays are sorted in ascending order.
- You are also given two numbers `m` and `n`, which represent the number of elements in arr1 and arr2 that are to be merged respectively.

Your goal is to merge two arrays into a single array in ascending order. The function should not return the final result array, instead should be stored inside of `arr1`. To accommodate this, `arr1` has a length of `m` + `n`, where the first `m` elements denote the elements that should be merged, and the last `n` elements are set to 0 and should be ignored. `arr2` has a length of `n`.

Constraints:

- *The length of arr1 is equal to m+n* *arr1.length == m+n*
- *The length of arr2 is equal to n* *arr2.length == n*
- *0 <= m, n <= 200*
- *1 <= m+n <= 200*

Examples:

```
Input: arr1 = [1,2,3,0,0,0], m = 3, arr2 = [2,5,6], n = 3

Output: [1,2,2,3,5,6]

Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6]with the underlined elements coming from arr1.
```

```
Input: arr1 = [12,20,33,70], m = 2, arr2 = [8,9], n = 2

Output: [8,9,12,20]

Explanation: The arrays we are merging are [12,20] and [8,9].
The result of the merge is [8,9,12,20]with the underlined elements coming
from arr1
```

```
Input: arr1 = [1], m = 1, arr2 = [], n = 0

Output: [1]

Explanation: The arrays we are merging are [1] and []. The result of the
merge is [1].
```

```
Input: arr1 = [0], m = 0, arr2 = [1], n = 1

Output: [1]

Explanation: The arrays we are merging are [] and [1]. The result of the
merge is [1]. Note that because m = 0, there are no elements in arr1. The
0 is only there to ensure the merge result can fit in arr1.
```

Requirements

- You must submit a single Java file containing the class named **MergeArrays**. You class must contain the function named "`merge`" with the following header:

```
public void merge(int[] arr1, int m, int[] arr2, int n)
```

Submission

- Download the source code for your implementation from Github Classroom. If you don't have a Github account, create one via github.com > Sign up.
- Select your name from the list of names available, which will link your GitHub ID. If you do not see your name on the list, speak with the instructor or one of the teaching assistants.
- Submit the repository link on Canvas. You may also add any comments in a README file (text, PDF or Word document) to help the grader understand or execute your implementation.

Grading

Your grade for this assignment will be determined as follows:

- 75% = Implementation: your class implementations must run successfully with the source files and data provided. It must produce the expected results, a sample of which appears in the Implementation section above. Any deviation from the expected results results in 0 credit for implementation.
- 15% = Decomposition: in the eyes of the grader, the degree to which your solution follows the suggestions above or otherwise represents a reasonable object-oriented and procedural decomposition to this problem.
- 10% = Style: your code must be readable to the point of being self-documenting; in other words, it must have consistent comments describing the purpose of each class and the purpose of each function within a class. Names for variables and functions must be descriptive, and any code which is not straightforward or is in any way difficult to understand must be described with comments.