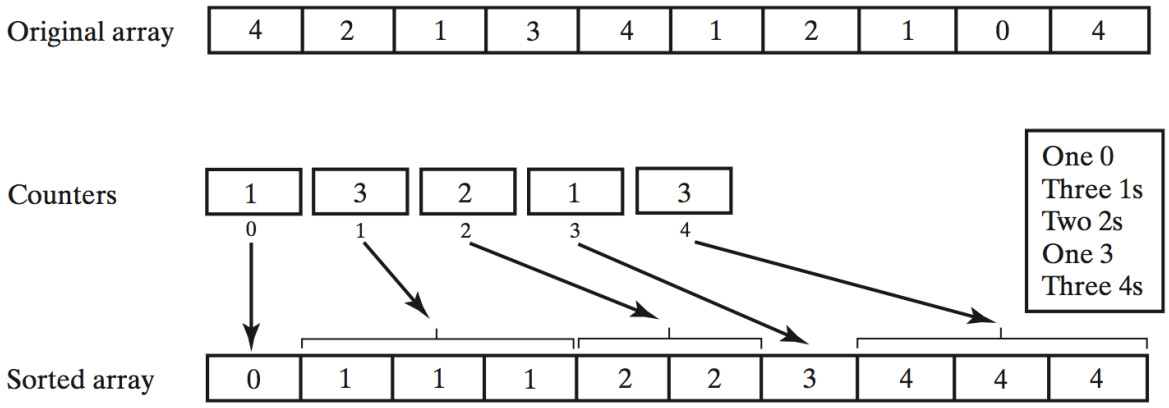
CMSC 256 – Project 3

Programming Assignment 3

Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor).

One way to sort an array of positive integers that lie between 0 and m, inclusive, is to count the number of times each value occurs in the array and store that information in counter variables. For example, the following image shows an array of integers that lie between 0 and 4 and the five counters after a counting sort has made its pass through the array.



The Counters record that the original array contains one 0, three 1s, two 2s, one 3, and three 4s. These are then interpreted by another method to build the sorted array.

Write a class named **`Sorter.java`** to sort integers in ascending order (from smallest to largest), using counter variables as described above. It should contain a class method, called **`sortIntArray`**, that accepts an integer array as its parameter and returns the array in sorted order. In addition, write a JUnit test class, called **`SorterTest.java`** that thoroughly tests your **`Sorter`** class.

Write this program in JAVA. Follow all commenting conventions discussed in class, including a comment block **at the top of each file** with your name, date, the course number and section, program purpose, etc. It is expected that your program will be well documented.

You will submit both the project files to Blackboard. Do not compress these files, but include both of them as a single submission.

In the comment section of your submission, answer the following questions:

1. Using Big Oh notation, describe the efficiency of this algorithm.
2. How does the efficiency of a counting sort compare to that of an insertion sort?
3. Is this algorithm useful as a general sorting algorithm? Explain why or why not.