QA-Code Challenge

Ivana Marjanovic

Intro

Array of integers is sorted out from smallest to largest number. This is done by using Counting sort algorithm (I find more about it here: https://www.programiz.com/dsa/counting-sort).

Bugs

About the bugs in this code, I find following problems:

```
import java.util.Arrays; //if it is not used we shouldn't import it

class JobFair {
  void doSomething(int array[], int size) { // Array contains non-negative integers
    int[] output = new int[size];        //variable size not checked for
                                       //negative value, and also if it is real
                                       //size of array
    // Find the largest element of the array
    int max = array[0];
    for (int i = 1; i < size; i++) {
      if (array[i] > max)
        max = array[i++]; //double post-incrementing i after new max is found,
    }                     //jump from i to i+2 (i+1 can never be reached)
                          //so max=array[i];
    int[] count = new int[max]; //not just max, it should be max+1
                   //array is: 1,2,3 so max is 3, but then array count only
                   //have entries 0,1,2 so when we try to reach count[max] it
                   //throws exception

    // Initialize count array with all zeros.
    int k = 0;
    while (k < max) { //k is never incremented so this makes infinity loop
                      //also there is no need for this while loop because in
      count[k] = 0;  //java when we initialize int array default value of every
                   //element is zero
    }

    // Store the count of each element
    for (int i = 0; i > size; i++) { //it should be i<size because, this way
      count[array[i]]++;             //part inside of for loop would never
                                    //be reached
    }

    // Store the cumulative count of each array
    for (int i = 1; i <= max; i++) {
      count[i] += count[--i];  //here we have pre-increment, i only has 0 and 1
                               //values here
    }                          //instead of count[--i] we should just do
                               //this count[i-1]
    // Find the index of each element of the original array in count array, and
    // place the elements in output array
    for (int i = size - 1; i >= 0; i--) {
      output[count[array[i]]] = array[i];//in array output max entry value is at
       count[array[i]]--;              //size-1 (in total size entries)
       }                               //output[count[array[i]]-1]

// Copy elements into original array
    for (int i = 0; i < size; i++) {
      array[i] = output[i];
    }
  }
}
```

About algorithm

Now about algorithm, while loop should not exist so there we could save there some time. Apart from that algorithm is nicely done. But for sorting arrays in general we could use more efficient algorithms.

For example, if we have array like this 0,5,1,5,12500000, Counting sort algorithm is not optimal for sorting this array out: firstly, unnecessary space is used (lot of it), and then in for loop (for cumulative count of each array) for only 5 numbers we have 12500000 passes through the loop. We should think on a bigger scale and how much time this function would be called like this. So instead of this algorithm we could use something like bubble sort, quicksort (one of the best) and so on.. (more could be find here https://www.geeksforgeeks.org/sorting-algorithms/).

Testing

I think it would be best if we use parameterized test (with `CsvSource`). That way we are able to run single test multiple times with different parameters. That way in input we could have different arrays that we can test this code out and see if it works.

Array examples:

1,2,3,4,5,6

10,3,5,3,3,5,6,10

0,3,3,3,0,0,0

1,1,1,1,1,1

255,3,1,25,1,1,1500,5,23,2

0

Empty array

For size, we just put size of array and with some of tests we could mess up size so we can see how code will perform.