CMSC 401 – Fall 2023

Programming Assignment #1

(due Sun, 9/10 - 11:59pm)

Dr. Eyuphan Bulut

CMSC 401- Algorithm Analysis with Advanced Data Structures

VCU College of Engineering

Assignment 1

- Implement a variant of Majority Element which looks for elements occurring more than N/3 times
  - Input:
    - array of N positive integers (size of the problem is N)
  - Output:

    1) The numbers (separated by space) occurring more than N/3 times, if exists (there could be at most 2 such numbers, so print smaller first)

    2) **-1,** if there is no such number in the array

- Input should be read into an array of integers: int[] (do not use ArrayList)
- The code should work on that array, without re-formatting the data e.g., into a linked list or any other data structure
- The algorithm should have **O(N) time complexity**
- Use of any Java built-in functions/classes is NOT allowed
  - With the exception of functions from Scanner, System.in and System.out (or equivalent) for input/output

Input-output formats

- Input Format:
  - First line: a single integer number N>=3, N<=1,000,000, showing the number of integers in the array (it is not in the array)
  - **Following N lines:** each contains a single positive integer containing the elements of the array
    - Each integer will be <=1,000,000,000
  - Input will always be **correct** with respect to the specification above (error handling is NOT needed)

Input 1:

```
7
3
100
100
3
100
3
2
```

Input 2:

```
5
67
2
1
67
5
```

Input 3:

```
4
10000
2
868
3
```

- Output format:
  - A single line:
    - **-1**, if the input array has no such element
    - **X**, if X is the only such element
    - **X Y**, if there are two such elements where X <= Y

Output 1:

```
3 100
```

Output 2:

```
67
```

Output 3:

```
-1
```

Hints

- Check Algorithm V from Lecture 2 slides
- Check how **Boyer & Moore approach** is iterating using a counter over the array
- Consider keeping two counters for two different candidates
- Decrease their counters if a third different element is seen (i.e., like removal of three different elements)
- Once entire array is processed, check the actual counts of candidates (there could be at most 2 candidates) and compare with N/3
- Print according to the format
  - If more than one element with count > N/3, print smaller first
- If you need clarification, email and ask (do not assume)
- The code will get only one input and provide one output. For multiple inputs, we will run it multiple times.
- Use Discord to discuss with others and me etc.

Input/output in Java

- Use Standard I/O to read input and write the result
- For Java, input: System.in, output: System.out
- **"Do Not"s**
  - Do not read from a disk file/write to disk file
  - Do not write anything to screen except the result
    - Ex: Human centric messages ("the result is", "please enter..") – you will lose points if you do so
    - Automated grading via script will be used for checking correctness of your output

Submission

- Date due: Sun, Sept 10th, 11:59 pm

- Submission through Canvas
  - Just submit the single Java source code file CMSC401_A1.java
    - **No need to zip. Don't worry about "-1", "-2" added to your file by Canvas for new versions.**
    - The file should have *your name* in a comment in the first line
    - Remember: in Java, class name should match the file name, and is case sensitive

- Please do NOT create your own packages
- Do NOT place the file into a folder – just submit the java file.
- Use standard I/O to read input (System.in, System.out) and output
- Make sure the **program compiles and WORKS!**
- Late submissions are accepted up to 2 days **only** with penalties!