Programming

1D Arrays Homework 2

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Problem #1: Find the Three Minimum Values

- Read a non-negative integer n(>= 3), then read in n integers. Find the 3 smallest numbers
  - Don’t change the array content
  - Don’t iterate on the array more than once
  - Make sure they’re printed out in order, from smallest to largest
- Input ⇒ Output
  - 5 **4 1 3 10 8** ⇒ 1 3 4
  - 3 **7 9 -2** ⇒ -2 7 9

Problem #2: Search for a Number

- Read in a non-negative N, then read N <= 200 integers [0 <= A[i] <= 500].
  - We will search in this array for numbers
- Then, read integer Q (for a number of queries), and read Q integers
  - For each integer, find the **last occurrence** of it within the array, and print out this index
  - If the doesn't exist within the array, print out -1 instead
- Input 5 **1 2 7 3 7** 3 **7 9 2**
  - Means an array of 5 numbers (1 2 7 3 7) and 3 queries (7 9 2)
- Output
  - 4 [7 exists in 2 positions (2 and 4). The last occurrence is 4]
  - -1 [9 doesn't exist within the array]
  - 1 [2 exists only in position 1]
- Do it first with nested loops. Can you do it **without** using nested loops?

Problem #3: Find the Most Frequent Number

- Read in a non-negative N, then read N <= 200 integers. Find the value that is repeated the most number of times.
  - Each integer is **-500** <= value <= 270
- Example for array: 7 **-1 2 -1 3 -1 5 5**
  - -1 is repeated 3 times: the most frequent number in the array
- Don’t use nested loops
- You can assume there is always a unique answer
  - Consequently, an array such as [1, 1, 2, 2], which has 2 valid answers, is invalid
    - Alternatively you can just print out either solution

Problem #4: Digit Frequency

- Read in a non-negative integer N, and then read N <= 200 integers. For every digit from 0 to 9, we want to know how many times each digit appears
  - Input 2 307 78
  - Output:
  - 0 1
  - 1 0 [digit 1 never appeared]
  - 2 0
  - 3 1
  - 4 0
  - 5 0
  - 6 0
  - 7 2 [digit 7 appeared twice]
  - 8 1
  - 9 0

Problem #5: Unique Numbers of an Unordered List

- Read **a** non-negative integer N (<= 900), followed by reading N integers (0 <= value <= 500)
- Print the **unique** list of the numbers, but **preserve** the given order
- Input: 13 **1 5 5 2 5 7 2 3 3 3 5 2 7**
- Output: 1 5 2 7 3
  - Observe: the input is **not necessarily a sorted** list
  - Observe: the output preserves the original order: i.e. 5 appears before 2 in this instance
- Don’t use nested loops

Problem #6: Sorting Numbers

- Read in **non-negative** integer N (<= 900), followed by N integers (0 <= value <= 500)
- Print out the numbers, completely **sorted** from small to large
- Input: 13 **1 5 5 2 5 7 2 3 3 3 5 2 7**
- Output: 1 2 2 2 3 3 3 5 5 5 5 7 7
- Give it your best attempt, and try to be efficient with your code
  - You don’t need to google how to sort numbers
  - Hint: the maximum value in the array is 500

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."