

CSC 323 Algorithm Design and Analysis, Spring 2020

Instructor: Dr. Natarajan Meghanathan

Assignment 2 (Programming): Design and Implementation of a $\Theta(n)$ Algorithm to Simultaneously Determine the Largest and Second Largest Elements of an Array

Due by: Feb. 11th, 11.59 PM

In this assignment, you will design and implement a $\Theta(n)$ algorithm that will simultaneously find the largest and second largest elements (integers) in an array. Note that the second largest element should be distinctly smaller than the largest element. You should also adhere to the following restrictions.

- (1) You should traverse through the array ONLY ONCE.
- (2) The array could have both positive and negative elements. So, you should NOT initialize any temporary variables to very small negative values as part of your algorithm.
- (3) You should NOT make any assumption regarding the values of the elements and their distribution in the array.

Note: Failure to adhere to the above restrictions and/or designing an algorithm whose time complexity is greater than $\Theta(n)$ will result in a score of ZERO.

Submission (Canvas)

Submit items 1, 2, 3, 4 and 6 together as one PDF file

Submit the C++ code (item 5) as a separate file

- 1) (10 pts) A pseudo code of the algorithm
- 2) (10 pts) Explain the working of the algorithm on your own in at least 150 words.
- 3) (10 pts) Analyze the time complexity of the algorithm and show that it is $\Theta(n)$.
- 4) (15 pts) Show the working of the algorithm with the following two types of example arrays:
 - (a) An example array of 10 integers that need to have the largest integer appearing at least twice.
 - (b) An example array with the largest integer and second largest integer as respectively the first and last element in the array.
- 5) (50 pts) A C++ code of the algorithm with the array (of size 10 and range: 1 to 50) randomly generated using the `srand(NULL)` and `rand() % maxValue` functions.
- 6) (5 pts) Two screenshots of the execution of your C++ code.