CSC 228 Data Structures and Algorithms, Fall 2017
Instructor: Dr. Natarajan Meghanathan

Project 3: Finding the Next Greater Element of an Element in an Array in Θ(n) Time

Due: October 4, 1 PM

In this project, you will design and implement an algorithm to determine the next greater element of an element in an array in Θ(n) time, where 'n' is the number of elements in the array. You could use the Stack ADT for this purpose.

The next greater element (NGE) for an element at index i in an array A is the element that occurs at index j (i < j) such that A[i] < A[j] and A[i] ≥ A[k] for all k ∈ [i+1..., j-1]. That is, index j (j > i) is the first index for which A[j] > A[i]. If no such index j exists for an element at index i, then the NGE for the element A[i] is considered to be -1.

For example: if an array is {1, 15, 26, 5, 20, 17, 36, 28}, then the next greater element (NGE) of the elements in the array are as follows:

1	15
15	26
26	36
5	20
20	36
17	36
36	-1
28	-1

Note: Your code need not print the elements and their NGE values in the same order of the appearance of the elements in the array. An out of order print is also acceptable as long as the correct NGE for an element is printed out. For example, the following out of order print for the above array is also acceptable.

1	15
15	26
5	20
17	36
20	36
26	36
28	-1
36	-1

You are given the doubly linked list-based implementation code for Stack along with this project description. Note that the main function in the code given to you already has the code snippet to generate an array of random elements. You just extend the main function to implement the algorithm.

Submission (through Canvas):
(1) Write a pseudo code of the algorithm designed and implemented for this problem.
(2) Discuss the time complexity of the algorithm and justify that it is Θ(n) with respect to the number of pop operations.
(3) Include the complete C++ or Java code of the Doubly Linked List-based Stack and the main function that includes the implementation of the algorithm.
(4) Show a screenshot of the execution of your algorithm for an array of 10 elements whose maximum value could be 50.