American University of Armenia, CSE
CS121 Data Structures B
Spring 2022

Homework Assignment 3

Due Date: Sunday, March 6 by 23:59 electronically on Moodle

Please solve the programming tasks using Java, following good coding practices (details are on Moodle). Don't forget to submit the corresponding assignment report.

1. **(10 points)** Write a generic recursive method for reversing the contents of a queue without using any additional data structure. Specify the time and memory complexity of your method. Test it in a program.

2. **(15 points)** Write a program that reads an input message consisting of lower-case Latin letters, finds and removes all pairs of similar neighbors and prints the remaining letters in their original order. You should use an appropriate data structure to complete this task.

sample input	sample output
`ccdatttaffstrrrrructwwurezzsss`	`datastructures`

3. **(15 points)** Implement the Queue ADT using an array as the underlying data structure such that the front of the queue always corresponds to index $n - 1$ where n is the size of the array. Write a program to test all the methods of your class. Specify and discuss the execution times of each of the Queue ADT methods in this implementation.

4. **(25 points)** Implement the Deque ADT using two stacks S and T. Your `StackDeque` class should implement the `Deque` interface given in the textbook. Write a program to test all the methods of your class. Specify and discuss the execution times of each of the Deque ADT methods in this implementation.

5. **(15 points)** Implement the Stack ADT using a *dynamic array*. All methods should run in $O(1)$ amortized time. The array capacity should be halved whenever the number of actual element falls below one-fourth of that capacity, and should be doubled when the number of actual elements overflows the capacity. Write a program to test all the methods of your class.

6. **(15 points)** Implement in-place recursive *Quick-Sort* using an array list to represent the sequence to be sorted and used for the intermediate steps of the algorithm. Use the first element as the pivot. Illustrate the use of your implementation in a program.