Name: Osei Owusu Ansah
ID: 10990203
Date: 05/06/2023

Assignment Submission

Overview: With an emphasis on the stack data structure, this report covers the implementation and analysis of several data structures and algorithms as assignments. The assignments consists of multiple exercises and implementations that are intended to strengthen my knowledge of stacks and their real-world uses.

Implementation of ArrayStack The ArrayStack class, a concrete realisation of the StackADT interface, is implemented in the ArrayStack.java file. The ArrayStack class makes use of an array to hold the stack's elements, guaranteeing effective memory use and constant-time access to the top element.

Standard stack operations, such as push, pop, peek, isEmpty, and size, are provided by the class. It also has an expandCapacity method that dynamically resizes the underlying array as the stack fills up, and a toString method for easy stack representation.and an expandCapacity method to dynamically resize the underlying array when the stack becomes full.

Largest Value in Array The FindLargest.java file implements an algorithm to find the largest value in an array. The findLargestValue method takes an array of integers as input and returns the largest value present in the array. The algorithm iterates through the array, comparing each element with the current largest value and updating the largest value accordingly.

Sentence Reversal The Reverser.java file contains an implementation that reads a sentence from the user and prints the sentence with the characters of each word reversed. It utilizes a stack data structure to reverse the characters of each word efficiently.

Time Complexity Analysis The time complexity of the findLargestValue algorithm implemented in FindLargest.java is O(n), where n is the number of elements in the array. This is because the algorithm iterates through the array once, performing a constant-time operation for each element.

The time complexity of the sentence reversal algorithm implemented in Reverser.java is also O(n), where n is the length of the input sentence. This is because the algorithm iterates through the characters of the sentence once, performing constant-time operations for each character.

Name: Osei Owusu Ansah
ID: 10990203
Date: 05/06/2023

Challenges and Overcoming Them While working on this assignment, several concepts posed challenges initially. Knowing the fundamentals of the stack data structure and its Last-In-First-Out (LIFO) concept was one of the biggest obstacles. I studied "Java Foundations" for a while to better understand the principles and get past this obstacle.
Ensuring the accuracy of the ArrayStack class during implementation was another difficulty. I had trouble at first with edge circumstances, like managing a full or empty stack. I overcome this difficulty by creating in-depth test cases and extensively evaluating the implementation, which enabled me to find and fix any problems. The sentence reversal exercise also needed careful consideration to make sure the algorithm functioned correctly for a range of input conditions. I broke down the problem into smaller steps and tested each step individually, which helped me identify and resolve any issues in the implementation. **Accessing the Code** The complete code for this assignment is available on my GitHub repository. You can access the repository at the following URL: https://github.com/PapaDaCodr/Stacks-Implementations.git. Feel free to explore the code, run the implementations, and provide any feedback or suggestions.

Conclusion This assignment has provided a comprehensive understanding of the stack data structure, its implementation, and its practical applications. By implementing the ArrayStack class and utilizing it in various scenarios, such as finding the largest value in an array and reversing the characters of words in a sentence, we have reinforced our knowledge of data structures and algorithms.

The time complexity analysis highlights the efficiency of the implemented algorithms, ensuring optimal performance for the given tasks.

Overall, this assignment has been a valuable exercise in understanding and applying data structures and algorithms, particularly the stack data structure, in real-world scenarios. The challenges faced during the assignment have helped strengthen my problem-solving skills and deepen my understanding of the concepts involved.