

Lab 9 Java Sorting & Regex

UWYO COSC 2030

1 Stacks, Queues, Deques, and Vectors in Java

Stacks, queues, deques, and vectors are data structures you have encountered in C++. This lab covers the same data structures in Java for familiar tasks, including string reversal, checking parentheses, and sorting. Links to the Java documentation for each of these data structures is included below:

- <https://docs.oracle.com/javase/8/docs/api/java/util/Stack.html>
- <https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>
- <https://docs.oracle.com/javase/8/docs/api/java/util/Vector.html>
- <https://docs.oracle.com/javase/8/docs/api/java/util/Deque.html>

Once you have completed these familiar tasks, you will then get started with regular expressions (regex) in Java.

2 Java String Reversal and Parentheses Checking

Complete the functions `stringReverse` and `parenCheck`. String reversal will use a stack and checking parentheses will use a queue. The functions `stringReverseVector` and `parenCheckDeque` will require you to use a deque and a vector, respectively.

2.1 String Reversal

Stacks use the “last-in first-out” style for storing data. Pushing the characters 'h','e','l','l','o' into a stack will yield 'o', 'l', 'l', 'e', 'h' (reverse order) upon using the pop operation on all elements in the stack.

2.2 Parentheses Checking

For the `parenCheck` function, you will check sets of parenthesis to ensure they are matched properly. Each time a '(' appears in the string, insert it into the queue. Use the dequeue operation when you get a matching ')'. For a perfectly matched string, you should have an empty queue by the end. Consider: what happens when you find a ')' with no matching '(' to pop?

3 Sorting

You will finish implementing the same basic sorting algorithms that you have seen before in previous labs. Quicksort is already completed and is included for your reference. Mergesort and Heapsort will need to be completed and include recursive calls.

4 Regular Expressions

Lecture slides and this link: https://www.w3schools.com/java/java_regex.asp should provide some background on using regular expressions in Java. Using regex, you will create three functions to evaluate and then accept or reject a variety of strings based on whether or not certain patterns are matched. The match conditions are as follows:

- If the input string has at least one occurrence of the substring 'Cowboys'

- If the input string is a 5-digit string starting with '7'
- Your own regex pattern – leave a comment above your function to briefly explain what the rule is, and it will be evaluated as described. Examples are helpful to clarify in case of ambiguity or uncertainty

5 Assignment

Accept the assignment here: <https://classroom.github.com/a/5TyR5HE1>. Complete the tasks described above.

6 Submission

Submit on Github Classroom. Remember to include a README.md file with your name and any assistance received.