BBM104: Introduction to Programming Laboratory II (Spring 2020)

Assignment 4 Report

Stack and Queue Operations

Tuana Çetinkaya - 21985164
7-6-2020

1. Problem Definition:

We are expected to implement a solution for Stack and Queue data structures to perform some specific operations. Such as add, remove, remove all greater numbers than some selected number etc.

2. Steps of Algorithms:

System first takes the initial stack and queue from text file and initialize them. Then it reads the command.txt and perform the operations in runtime and print the result to output files after every operation. It constantly checks and updates the size of both structures to avoid errors. Finally it updates the original stack and queue files to their latest version.

3. Analysis for “removeBiggerThan” for Stack:

Time complexity of that method is O(n) for Stack.

This method uses a helper stack to keep the content safe. Let’s assume one line of code takes t time (in case it is so fast that we will ignore the most of it to get to the complexity).

Method first creates a stack and keeps the initial size and check if the stack is empty. The whole process takes roughly 4*t time.

Then inside a for loop, method checks if the top value is smaller or equal to the given number, if so it pushes the value to helper stack to keep it then pop it from the stack. If not, it just pops the element to get rid of it. This if-else block takes 2*t time but since it’s inside a for loop that happens for number of elements inside the list it happens 2*n*t time.

Then it initializes another size to keep the new length of the stack (1*t time) and pushes the elements back to stack inside a for loop that occurs in n*t time.

Total time complexity is roughly:

$(4 + 2n + 1 + n)t = (5 + 3n)t$

We can ignore the additional 5t since it’s extremely short interval of time, and we’ll be left with 3nt complexity. So “removeBiggerThan” method has the complexity of O(n) for this case.