

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 4 REPORT

DENİZ CAN ERDEM YILMAZ
151044001

6. Test Cases

There isn't too much things to test due to lack of input number and Null reference errors. But all errors might come from non instantiated variables are handled by creating new instances. File checkers control the existance of file and situation of read/write operations. Any error due to empty file, no file, file open errors can be caught by try-catch blocks. This includes input mismatches from taking inputs from file. In any abnormal situation program gives appropriate error message and quits.

7. Running and Results

Q1.

There were 4 different stacks with using same data set. Timers on each test nearly same and changes only a little bit with each time. Analayze can be done by the rate of times on each run. Stack with arrayList as container is the fastest one. Extending arrayList and containing a node to store elements takes nearly same time and takes second place on process speed. Appearently stroing datas in a stack that contains queue as data container which extends a linked list is not a good idea for performance issues can seen on placement. Runing without a test.csv cause an error just like running mixed type of datas on a line in test.csv. Due to creating 4 different stacks output file is 4 times bigger than input file, includes size of lines on first column and is reversed while writing to file with pop order.

Q2.

There are 2 different reverse methods, one is inside the MyQueue class and the other is in TestClass as a static method. Each methods takes MyQueue objects to reverse the elements in queue. While the member method does without creating new node, static one does it with recursive algorithm. Both does their jobs well (can be seen on result file) but recursive one seems faster than the no-new-node one. This might change on bigger data sets

Q3.

Timers and tests are done for only offer and poll methods. From timers queue which contains linkedList can declared as faster than the one extends it. No other runing differences has found. Result file's size has doubled due to double queue. deleteMin() method has implemented differently from trivial queue classes. It finds the element that has the minimum value of data set and deletes it. Seems like it crosses the borders of the fundamentals of being a queue but this was requested as homework.

Screenshots can be found of tests and results inside the SS folder