Homework Set 1

Chongyi Xu

April 5, 2017

Write Up Part 1

1. Why did you choose your particular tests in the `QueueTest.java` file? For `testEmpty` and `testOne`, a couple sentences will do. For `testMany`, explain why you think some implementations might fail those tests.

   I test `front()` and `dequeue()` in `testEmpty` by call the methods with an empty queue. In this way, I can see if it successfully returns null to me. And I just call `testOne` to test `enqueue()`.
   As for `testOne`, I test the case with one element in the queue, and see if the implementations of my queue and java queue are the same.
   `testMany` might fail the tests if `dequeue()` does not work appropriately with empty queue(when `dequeue()` to an empty queue).

```
testEmpty(ListQueue yourQueue, JavaQueue correctQueue){
    if (front() returns null and dequeue() returns null) {
        test enqueue() use testOne()
        if pass, return true
    }
    otherwise, return false
}

testOne(ListQueue a, JavaQueue b){
    assign a and b with one element
    if (front() works as JavaQueue does) {
        test enqueue()
        if (dequeue() works as JavaQueue does) {
            return true
        }
    }
    otherwise, return false
}

testMany(ListQueue a, JavaQueue b){
    create stack s1, s2 to store strings that will be enqueued
    s1 has strings a, b, ..., k, 11 in total
    s2 has strings abcdefghljk, bcdefghljk, ..., k, 11 in total
    use private method to test s1, s2
    testMany(ListQueue a, JavaQueue b, Stack<String> s) {
        enqueue all strings in the stack
        if front() is different from JavaQueue, return false
```

```
            dequeue all strings in the Queue
            if dequeue() is different from JavaQueue, return false
            return true if front() is null after all strings are dequeued
        }
        if both s1 and s2 pass test, return true
        return false
    }
```

2. After running your tests on your code, how confident are you that your implementation is correct? Explain why you think your test cases are sufficient, or alternately explain what additional tests might be prudent. These explanations should be at a high level and do not require any implementation.

   I am quite confident that my implementation is correct. Because I have tested empty case(special case), one-element case(base case), and many-element case(inductive case). And in the case of many elements, I used more than 10 elements in the queue to make sure the case is representative enough.