UNIVERSITY OF VIRGINIA, DEPARTMENT OF COMPUTER SCIENCE

Concurrency - Concurrent Queue Implementation

Nada Basit and Mark Floryan

March 9, 2022

1 SUMMARY

For this homework, you will be implementing a simple concurrent queue. For this assignment, we are going to focus on correctness (i.e., the queue must work with multiple threads without crashing). Optimizing the speed of your concurrent queue is recommended, but ultimately optional.

1. Download the provided starter code
2. Implement the ConcurrentQueue class
3. Run the main in MainTester.java to ensure your queue works correctly
4. **FILES TO DOWNLOAD:** concurrency.zip
5. **FILES TO SUBMIT:** ConcurrentQueue.java

1.1 CONCURRENTQUEUE.JAVA

To begin, implement the *ConcurrentQueue* class inside the ConcurrentQueue.java file. The methods you are responsible for are listed below. This Queue **must be a linked-list based queue**. You may use Java's built-in Linked List (import java.util.LinkedList) or you may use your own implementation from the previous homeworks. *You may NOT import Java's built-in Queues.*

```
1   public class Queue<T>{
        public Queue();
3
        public int size();
5
        public void enqueue(T data);
7
        public T dequeue();
9   }
```

Your class must be alterable by different threads. Thus, you should protect the fields (head, tail, etc.) of your linked list from being corrupted by multiple threads accessing the queue at once. For this homework, we care about *correctness* (i.e., the queue works as intended with multiple threads even if it could be optimized to run faster).

1.2 MAINTESTER.JAVA

You can test your code by running the main method in *MainTester.java*. This tester will first test your queue using a single thread and time the results. Then, the method will test your queue again using two threads and time the results again. Any errors that occur should be printed to the console. Make sure you run the tester multiple times. Race conditions can sometimes cause code to appear to work but not consistently.

1.3 OPTIONAL: OPTIMIZING

Although we only care about correctness for this homework, if you are interested you should try to make your queue run as quickly as possible. What can you do to increase the degree to which multiple threads can use the queue in parallel? How fast can you make the concurrent test that we provided?

When you are done, submit your entire project as a zip file to Collab.

1.4 GRADESCOPE

You should submit your code to *Gradescope*. If you are having trouble with your submission, you should double check the following common problems:

1. Make sure you are only submitting one file, and it is called *ConcurrentQueue.java* exactly.

2. Make sure you keep the package statement at the top of the file.

3. Your *ConcurrentQueue.java* should **NOT contain any imports of any of Java's built-in Queue data structures**.

4. Make sure your file is **NOT printing anything extra to the console**.