Week 8 worksheet: Stacks, Elementary Search, Queues II

Total points: 10

Out: March 25 (Tuesday)

Due: March 31 (Monday end of day [2359 CDT according to D2L])

## What to submit?

Upload only one Word or PDF file to the designated ***Worksheet 9*** D2L folder.

# Exercise 1: Queue Implementation Using a Circular Array

[5 pts] Consider the queue implementation using a circular array. Show the contents of the **data** array and the values of **front** and **rear** instance variables after executing each line of the following code. I’ve done the first one for you. I declared it as a java.util.concurrent ArrayBlockingQueue, but that doesn’t matter.

**ArrayBlockingQueue<Integer> abq = new ArrayBlockingQueue<Integer>( 5 );**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **operation** | **index 0**  **value** | **index 1**  **value** | **index 2**  **value** | **index 3**  **value** | **index 4**  **value** | **front** | **rear** |
| abq.add(100) | 100 | ∅ | ∅ | ∅ | ∅ | 0 | 0 |
| abq.add(200) | 100 | 200 |  |  |  |  |  |
| abq.add(300) | 100 | 200 | 300 |  |  |  |  |
| abq.remove() | 200 | 300 |  |  |  |  |  |
| abq.add(400) | 200 | 300 | 400 |  |  |  |  |
| abq.add(500) | 200 | 300 | 400 | 500 |  |  |  |
| abq.add(600) | 200 | 300 | 400 | 500 | 600 |  |  |
| abq.remove() | 300 | 400 | 500 | 600 |  |  |  |
| abq.add(700) | 300 | 400 | 500 | 600 | 700 |  |  |

# Exercise 2: Move to front of queue

[5 pts] Write a method that takes a generic java.util.Queue<E> and an item of type E, and returns a Queue<E> identical to the original queue, but with all occurrences of *item* moved to the front of the queue. You may use any method of the java.util.LinkedList and java.util.Stack classes and java.util.Queue interface. You may assume that there is an equals method for type E. You must preserve the original queue. For example, using Integers with item = 7,

queue | 8 | 5 | 7 | 4 | 7 | 5 | 9 | 7 |

front rear

becomes | 7 | 7 | 7 | 8 | 5 | 4 | 5 | 9 |

front rear

The signature of this method is

**public static Queue<E> moveToFront( Queue<E> q, E item )** **{**

**// Create a new queue to hold the result**

**Queue<E> resultQueue = new LinkedList<>();**

**// Create a temporary stack to hold non-matching items**

**Stack<E> tempStack = new Stack<>();**

**// Iterate through the original queue**

**while (!q.isEmpty()) {**

**E current = q.poll(); // Remove the head of the queue**

**if (current.equals(item)) {**

**// If the current item matches, add it to the result queue**

**resultQueue.add(current);**

**} else {**

**// Otherwise, push it onto the stack**

**tempStack.push(current);**

**}**

**}**

**// Now add the remaining items from the stack to the result queue**

**while (!tempStack.isEmpty()) {**

**resultQueue.add(tempStack.pop());**

**}**

**return resultQueue;**

**}**