

# COL106: Data Structures, I Semester 2015-16

## Assignment 1

### Stack Sortable Permutations

Deadline: **11:55PM, 7 September 2015**

August 27, 2015

The aim of this assignment is to test whether a given sequence of numbers is a **stack-sortable permutation** or not. A stack-sortable permutation is a sequence of numbers which can be sorted using a stack, and a queue. Kindly use this wiki page to understand what a stack-sortable permutation is: [https://en.wikipedia.org/wiki/Stack-sortable\\_permutation](https://en.wikipedia.org/wiki/Stack-sortable_permutation).

<b>Algorithm 1:</b> Algorithm: isStackSortablePermutation
<p><b>Data:</b> queue inputQ <b>Result:</b> true if input queue is a stack-sortable permutation, false otherwise</p> <pre>1 initialize an empty stack: s; 2 initialize an empty queue: outputQ; 3 <b>while</b> <i>inputQ is non-empty</i> <b>do</b> 4     element = input.dequeue(); 5     While the stack is nonempty and element is larger than the top        item on the stack, pop the stack, and enqueue it to output; 6     push element on stack; 7 <b>end</b> 8 <b>while</b> <i>s is non-empty</i> <b>do</b> 9     element = s.pop(); 10    enqueue element in outputQ; 11 <b>end</b> 12 return true if outputQ contains a sorted list of numbers;</pre>

Algorithm isStackSortablePermutation determines whether a given se-

quence of numbers is stack-sortable or not.

At the end of the algorithm, if the *output* queue contains a sorted list of numbers, return true.

In this assignment, you have to implement the `isStackSortablePermutation` method along with the stack, and the queue data-structures. The format of the assignment is similar to the “Java lab module”. The tar-ball for the assignment code can be downloaded from the course home page.

The tar-ball contains four files:

1. `checker.java`: This is the main checker file. It will call the stub functions that you write, and check whether the output that you report is correct or not. DO NOT make any modifications to this file.
2. `myqueue.java`: This file should contain the implementation of a queue. You must fill three stub functions in this file: `dequeue`, `enqueue`, and `isEmpty`. DO NOT modify the input/output parameters for this methods. You can add extra methods to this file, but you must maintain the semantics of the stub functions.
3. `mystack.java`: This file should contain the implementation of a stack. You must fill four stub functions in this file: `push`, `pop`, `isEmpty`, and `getElementAtTopOfStack`.
4. `test.java`: This file should contain the implementation of `isStackSortablePermutation` method. You must fill the stub function: `isStackSortablePermutation`.

Use arrays to implement the stack and the queue data-structure. For simplicity, assume that the maximum number of input elements is 100.

At the end of this assignment, you should understand the following concepts thoroughly:

1. Stack
2. Queue
3. Algorithm to sort with a stack

**Grading:** The grading of the assignment is divided in two parts: 30 marks (executing the code with all test cases), and 70 marks (understanding of concepts that you learnt in this assignment).

**Submission :** The deadline for the assignment submission is 7th September (11:50 PM). The submission should be a zip file which should contain the following files: checker.java, myqueue.java, mystack.java, test.java and Makefile. If you do not submit any code on Moodle till the submission deadline, you will get 0 marks. Also, note that a student is eligible to appear for demo ONLY if he/she has submitted *some* code on Moodle. Additionally note that any submission that is mailed to the TA/instructor will NOT be considered for grading purposes.

How to run the assignment:

```
$ .... Download assignment1 tar-ball .... $ tar -xvf assignment1.tar.gz
$ cd assignment1
..... Make suitable changes to myqueue.java, mystack.java, and mytest.java
.....
$ make
```

If the make command is not working, replace it with following commands:

```
$ javac myqueue.java
$ javac mystack.java
$ javac test.java
$ javac checker.java
$ java checker
```