

CS102 - Algorithms and Programming II
Lab Programming Assignment 7
Fall 2016

ATTENTION:

- Feel free to ask questions on Moodle on the Lab Assignment Forum.
- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:
CS102_SecX_Asgn7_YourSurname_YourName.zip
- Replace the variables “YourSurname” and “YourName” with your actual surname and name and X with your Section id (1, 2 or 3).
- Upload the above zip file to Moodle by the deadline before the lab (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TA during the lab. You will resubmit your code once you demo your work to the TA.

GRADING WARNING:

- Please read the grading criteria provided on Moodle.

Q1. Write a class that implements a queue structure by using link list. You should decide which links to be kept in link list for efficient queue implementation. Your queue should keep an integer inside of it and have the following functionalities:

- **class Node:** Inner class for keeping list items. This class has two variables; integer item and Node next
- **Queue():** Constructor that creates empty queue.
- **void enqueue(int item):** Add the item to the queue.
- **int dequeue() throws Exception:** Remove and return the item on the queue least recently added.
- **int peek() throws Exception:** Return the item least recently added to the queue
- **int size():** Returns the number of elements on queue.
- **boolean isEmpty():** Returns true if queue contains no elements

Q2. Write a class called **TopologicalSorter** that have the functionality of calculating topological order of given tasks. You have N jobs to process on a computer processor. Some of the jobs must complete before others can begin. Specifically, you are given a list of order pairs of jobs (i, j). Find a sequence of the jobs such that for each pair (i, j) job *i* is scheduled before job *j*. You can solve this problem as described in the following algorithm:

- Hold an array of Queues. Queue[] out. out[i] maintains a queue of jobs that must be scheduled after i (outgoing nodes for i).
- Hold an array of integers, int[] in. indegree[i] maintains the number of jobs that must be scheduled before i (indegree nodes for i).
- Create a new queue that will keep the nodes with zero incoming nodes. Those that must be scheduled first.
- Repeatedly remove a node with zero indegree, put it in output string. Since this job is done, decrease by 1 the indegree of all the nodes in out[i] .
- You can assume that given job pairs will be always sortable.
- Write a main class and test your TopologicalSorter class.

- **Example:**

Tasks:

0 3

3 5

5 7

7 4

4 2

2 1

1 6

Scheduled: 0 3 5 7 4 2 1 6