Homework 7

Due April 6th, 2021

11:59pm 100 points

CS 2235

Data Structures and Algorithms

1. In manufacturing, most factories operate as a queue using the FIFO approach. Orders are processed as they are received on a first come first serve basis. However, some companies allow for "rush jobs", where certain orders can jump to the front of the line. This usually comes at a great cost. In this assignment you will create a program to simulate this type of factory using an Order class that you will develop and a priority queue.
    a. Your program should run in a loop, each iteration of which corresponds to a day of work at the factory.
    b. Each order is assigned a priority, an integer between 1 (highest priority) and 20 (lowest priority). From among all orders waiting to be filled, the factory must work on completing the order with highest priority.
    c. Each job must has a name which is simply "Order_" + an integer beginning at 1 and increasing by 1, with each new order added to the queue.
    d. In this simulation, each order will also come with a length value, which is an integer between 10 and 50, indicating the number of days that are needed to process the order. For simplicity, you may assume an order is not interrupted, once it is has started production.
    e. As most factories are in business to make money, it is important to keep track how much revenue is generated by each order. Your Order class should also have a value equal to the length of the order, multiplied by 25000.
    f. Your simulator must output the name of each Order being produced in each day and its priority.
2. Begin your simulation with ten randomly simulated Orders. Randomly assign a priority and a length value, and add them to your priority queue.
    a. Every 50 days, add another Order to the priority queue.
    b. There should be a 50% chance that the order is a rush job. If this happens, ensure that the priority of this order is 1 and the value of the order is three times the normal rate.
    c. If the order is not a rush job, it goes to the back of priority queue.
    d. If the queue is empty before the simulation ends, then the company

must heavily discount a project to keep the factory running. Add a job of random length and priority with a value of 10000 per day.

3. The simulation ends after 2 years (730 days). Print the total number of Orders processed, number of rush jobs, number of discounted jobs and amount of revenue earned during the simulation.

Demonstrate that your program works. Submit your source code and output screenshots. Utilize good object-oriented programming and efficient algorithms.

HINT I suggest creating a class “Order” which has instance variables, such as “name”, “priority”, “value” and “length”, and any other necessary methods. Then, instantiate a PQ using the java.util.PriorityQueue class of these Orders. Also, it is important to note that the PriorityQueue package does not allow objects that are not comparable to be put in a priority queue. When you create your Order class, make sure to implement the Comparable interface from java.util.

Scoring:
20%- Use of object oriented programming.
20%- Each order has name, priority, value and length associated with it.
20%- Orders are properly processed until completed, based on priority.
20%- Program coded, so that new orders are added as orders are processed.
10%- Order number and priority printed for each time step.
10%- Meaningful comments and proper header.