Lab 3: Heap

Dong Li

This lab is designed to give you practice working with heap. This is an individual assignment; you may not share code with other students.

Specification

Your task is to solve the event simulation problem. Before that, you need to implement the priority queue with heap. The input stream consists of ordered pairs of arrival time and service time for each customer, sorted by arrival time. We use a quantum unit.

Customer data structure. To model a customer with arrival time and service time, create a data type Customer with the following API:

```
public class Customer {
    // create a Customer
    public Customer (int arrival, int service);
    // get the arrival time
    public int getArrival();
    // set the arrival time
    public void setArrival (int arrival);
    // get the service time
    public int getService();
    // set the service time
    public void setService(int service);
}
```

Binary Heap data structure. To model a binary heap, create a data type BinaryHeap with following API:

```
public class BinaryHeap {
    //create a BinaryHeap
    public BinaryHeap(int capacity);
    public BinaryHeap(int[] items);
    //insert a element into the heap
    public void insert(int x);
    //return the minimum element
    public int findMin() throws EmptyException;
    //remove and return the minimum element
    public int deleteMin() throws EmptyException;
    //judge whether the heap is empty
    public boolean isEmpty();
    //remove all elements
    public void makeEmpty();
}
```

Event simulator data structure. To model an event simulator, create a data type EventSimulator with the following API:

```
public class EventSimulator {
    //create an event simulator
    //tellers determine how many customers can be serviced at the same time
    //arrivalLine is the input stream, sorted by arrival time
    public EventSimulator(int tellers, ArrayList<Customer> arrivalLine);
    //set the arrival line
    public void setArrivalLine(ArrayList<Customer> arrivalLine);
    //start simulation
    public void simulate();
}
```

You can use `java.util.Queue` when you need. But you should implement the binary heap data structure by your own.

Test case

The following method is an example to test the data type you implement.

```
public static void main(String[] args) {
        ArrayList<Customer> arrivalLine = new ArrayList<Customer>();
        arrivalLine.add(new Customer(0, 8));
        arrivalLine.add(new Customer(1, 5));
        arrivalLine.add(new Customer(2, 3));
        arrivalLine.add(new Customer(4, 1));
        arrivalLine.add(new Customer(4, 5));
        arrivalLine.add(new Customer(4, 4));
        arrivalLine.add(new Customer(5, 4));
        arrivalLine.add(new Customer(6, 1));
        arrivalLine.add(new Customer(7, 3));

        EventSimulator es = new EventSimulator(3, arrivalLine);
        es.simulate();
}
```

The output is as follow:

```
Tick 0: process customer who arrival at tick 0 and leave at tick 8
Tick 1: process customer who arrival at tick 1 and leave at tick 6
Tick 2: process customer who arrival at tick 2 and leave at tick 5
Tick 5: process customer who arrival at tick 4 and leave at tick 6
Tick 6: process customer who arrival at tick 4 and leave at tick 11
Tick 6: process customer who arrival at tick 4 and leave at tick 10
Tick 8: process customer who arrival at tick 5 and leave at tick 12
Tick 10: process customer who arrival at tick 6 and leave at tick 11
Tick 11: process customer who arrival at tick 7 and leave at tick 14
```

Submission

Create a zip file named **YourStudentID.zip** that contains your code project and upload your zip file to FTP server.

Deadline

18 Oct 2017 24:00 GMT+08:00 (**TODAY !**)