Lab 12: Heaps

Professor: Ronaldo Menezes

TA: Ivan Bogun

Department of Computer Science
Florida Institute of Technology

November 4, 2013

1 General description

1.1 Legend

According to wikipedia[1] a heap is a specialized tree-based data structure that satisfies the heap property: If A is a parent node of B then the key of node A is ordered with respect to the key of node B with the same ordering applying across the heap. Heaps can be used to implement priority queue. In this lab you are asked to implement the heap using array which will be used to handle printer queue, where priority is the time where the document was send to the printer. That is, the document sent to the printer earlier should be printer before the document which was sent later.

2 Implementation

Implement the class *Heap.java*

```
// Heap.java
public class Heap {

    private PrintJob[] pq; // array of printer jobs
    private int N = 0;    // number of elements in the heap

    public Heap(int maxN) {
    // create the heap of size maxN
    }

    public void insert(PrintJob v) {
    // insert element v into the heap
    }

    public PrintJob deleteMinimum() {
    // delete the element with minimum priority from the Heap and return it.
    }

    private void swim(int k) {
    // swim the element with index k
    }

    private void sink(int k) {
    // sink the element with index k
    }
```

Use the following class *PrintJob.java*. Nothing to implement here.

```
// PrintJob.java
public class PrintJob {
```

[1] http://en.wikipedia.org/wiki/Heap_(data_structure)

```
    private String description;      // name of the document to be printed
    private Date date;

    public String getDescription() {
        return description;
    }

    public PrintJob(String description, Date date) {
        this.description = description;
        this.date = date;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    }
```

Implement the class *Date.java*, which will be used to denote the time where the document was sent to the printer.

```
// Date.java
public class Date implements Comparable<Date> {

    int hours;
    int minutes;

    public Date(int h, int m) {
        this.hours = h;
        this.minutes = m;
    }

    }
```

No imports are allowed in this lab.

3 Sample input-output

3.1 Input

Use the following main for testing.

```
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int N = 5;
        PrintJob pj;
        String res;

        Heap heap = new Heap(N);

        pj = new PrintJob("test.pdf", new Date(10, 11));
        heap.insert(pj);

        pj = new PrintJob("document.doc", new Date(10,8));
        heap.insert(pj);

        pj = new PrintJob("paper1.pdf", new Date(10, 15));
        heap.insert(pj);

        pj = new PrintJob("lab1.txt", new Date(11, 01));
        heap.insert(pj);

        pj = new PrintJob("lab2.txt", new Date(10, 6));
        heap.insert(pj);

        for (int i = 0; i < 5; i++) {
            pj = heap.deleteMinimum();
            System.out.println(pj.getDescription() + " " +
                pj.getDate().toString());
        }
    }
```

3.2 Output

```
lab2.txt  10:6
document.doc  10:8
test.pdf  10:11
paper1.pdf  10:15
lab1.txt  11:1
```

4 Grade breakdown

basis	grade
Implementation	(60)
Heap	50
Date	10
Comments	(20)
Javadocs	10
General	10
Overall	(20)
Compiled	5
Style	5
Runtime	10
Total	100