Assignment/Homework #3
COP-3530, Spring 2017

Rules & Instructions:

- **Due date: Monday, March 27, 2017 at 5 pm (Eastern Time)**
- This assignment has **4 problems**.
- The assignment/homework will be submitted **by email** to abajuelo@fiu.edu
- Your submission must be **a ZIP file** (not RAR format).
- **Please name your submission as 3_xxxxxxx.zip, where xxxxxxx is your seven digit Panther ID number)**.
- Please include the following header for each Java program:

```
/*************************************************************************
   Purpose/Description: <a brief description of the program>
   Author's Panther ID:  <your Panther ID number>
   Certification:
     I hereby certify that this work is my own and none of it is the work of
     any other person.
 *************************************************************************/
```

- Please indicate in the **subject of your email message** the following information:

COP-3530, SECTION U01, ASSIGNMENT #3

- Please submit all answers **ONLY as typed files** (using i.e. MS Word, WordPerfect, TextMaker, LaTex, etc). **Photos and scanned images will be not accepted**.
- Please make sure that you do not include any other personal information in your submission (besides the **Panther ID** in the name of the ZIP file and in the headers of your Java files as explained above). For example, no date of birth or name should be found in the document(s) you submit.
- Submissions turned in after the due date and/or which don't meet the established formatting rules will not be accepted.

Problem #1:

Consider a **binary (min)heap**. We have a method that prints the keys as encountered in a **preorder** traversal.

(a) Is the output sorted? Justify your answer.

(b) Attempt the same question for **inorder** and **postorder** traversal.

Problem #2:

(a) Consider an initially empty max-heap, where the following keys are to be inserted one at a time: 11, 19, 23, 12, 13, 17, 13, 14, 18, and 33. Draw the tree that results after building this max-heap.

(b) Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2, one at a time and in the giver order , into an initially empty **binary min heap**.

(c) Show the result of performing three **deleteMin** operations in the heap of the previous **binary min heap**.

Problem #3:

(a) Write a method **replaceKey** in the **MinHeap** class with the following signature:

public void replaceKey(Integer oldKey, Integer newKey)

The method will replace the first occurrence of **oldKey** with the **newKey**, and restore the **Min-Heap** property after the change. If the **oldKey** does not exist in the heap, the method prints an appropriate message and returns without changing the heap.

Example: Suppose our binary heap object (**bh**) has the following keys:

***	4	6	7	32	19	64	26	99	42	54	28

Then the method call: **bh.replaceKey (oldKey Integer(54), newKey Integer(2))** should change the keys to:

***	2	4	7	32	6	64	26	99	42	19	28

Note: You can assume that the methods **perlocateUp** and **perlocateDown** are already implemented in your **MinHeap** class.

(b) What is the **running time complexity** of your **replaceKey** method? *Justify.*

Problem #4:

Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function

h(key) = key mod 10,

show the resulting:

(a) Hash table using linear probing.

(b) Hash table using quadratic probing.

(c) Hash table with second hash function $\mathbf{h_2(key) = 7 - (key \bmod 7)}$.

(d) Consider two sets of integers, $S = \{s_1, s_2, ..., s_m\}$ and $T = \{t_1, t_2, ..., t_n\}$, $m \leq n$. Propose an algorithm (only pseudo-code) that uses a hash table of size m to test whether S is a subset of T.