Carleton University
SYSC 2100: Algorithms and Data Structures
Winter 2018

Assignment 5: Advanced Implementation of Tables
Due: Monday, April 9th 2018 at 11:55pm

This assignment consists of two parts:

1. The heapsort algorithm uses a heap to sort an array of items that are in no particular order. The strategy for heapsort is as follows:
   - Transforms the array into a heap
   - Removes the heap's root (the largest element) by exchanging it with the heap's last element
   - Transforms the resulting semiheap back into a heap

   The pseudocode is as follows:

```
+heapSort(in anArray:ArrayType, in n:integer)
    // build initial heap
   for (index = n-1 down to 0)  {
         heapRebuild(anArray, index, n)
   }
   //initialize the regions
   last = n-1

  for (step = 1 through n) {
     swap anArray[0] and anArray[last]
  //expand the Sorted region, shirnk the Heap region
     Decrement last
  //make the Heap region a heap again
     heapRebuild(anArray, 0 , last)
  }
```

   In the main program, use the following array as the test case for sorting into ascending order.

10	9	6	3	2	5

2. Implement the ADT table described in Chapter 13 of the text book using a **2-3 tree**. At minimum, the 2-3 tree should have the intended functionality for the following operations.
   - Inserting an item into 2-3 tree
   - Deleting an item from 2-3 tree
   - Traversing items in a 2-3 tree

   For simplicity, you can assume that the table only stores integers and that the key equals the value (i.e, you are implementing a set using a 2-3 tree).

   In the main line of the program provide test code that inserts `1 2 3 4 5 6 7 8 9 10`, prints the table, then deletes `3` and `7`, prints the table, and finally deletes `13`.

   Submission Requirements: Submit your assignment (the source files) using cuLearn. Your program should compile and run as is in the default lab environment, and the code should be well documented. Submit all files without using any archive or compression as separate files. The main program for part 1 should be called `TestHeapSort.java` and for part 2, your submission should be called `TestTwoThreeTree.java`. If you need to define additional classes etc., you are free to name them according to your own needs. But the TA(s) should be able to run your application by entering `java TestHeapSort` and `java TestTwoThreeTree` on a command-line.

Marks will be based on:

	25%	**50%**	**75%**	**100%**
Part 1(6 marks)	Runs	Sorts, but not exactly heapSort.	Sorting using heapSort.	Successful sorting using heapSort and handles errors..
Part 2 (10 marks)	Runs	Inserts and prints correctly.	Deletes items in list correctly.	Handles errors gracefully.
Readability (2 marks)	Hard to decipher.	Dead or useless code or variables, reinventing wheel, methods that do too much	Follows conventions, proper whitespace and indenting, appropriate and meaningful naming of variables and methods.	Includes logging or testing code for tree insert and delete (commented out or disabled for submission)
Comments (1 marks)	Occasional	Lots, but extraneous (i.e, `i=j // copy j to i`.)	Some	Javadoc comments for all member functions, classes. Minimal and cohesive comments elsewhere.
Completeness of your submission (1 marks)	Something in CULearn	TA must change code to get it to run.	RAR, JAR, ZIP files, incorrect naming.	Strictly adheres to the submission requirements.

The due date is based on the time of the cuLearn server and will be strictly enforced. If you are concerned about missing the deadline, here is a tip: multiple submissions are allowed. So you can always submit a (partial) solution early, and resubmit an improved solution later. This way, you will reduce the risk of running late, for whatever reason (slow computers/networks, unsynchronized clocks, failure of the Internet connection at home, etc.).

In cuLearn, you can manage the submission until the deadline, taking it back, deleting/adding files, etc, and resubmitting it. The system also provides online feedback whether you submitted something for an assignment. It may take a while to learn the submission process, so I would encourage you to experiment with it early and contact the TA(s) in case you have problems, as only assignments properly and timely submitted using cuLearn will be marked and will earn you assignment credits.