Faculty of Science

Course:	CSCI 3070U: Analysis and Design of Algorithms
Assignment:	#2
Topic:	Recurrences, heaps, heapsort, dynamic programming

Note: This assignment is to be completed individually. This means you should not work in teams. These restrictions are in place because the idea of this assignment is to prepare you to do well on the upcoming midterm, which will include questions related to this assignment.

Part 1

For this part of the assignment, you will implement the heap data structure, and use it to implement a heap sort in Java, C, C++, Python (or another approved programming language).

The heap data structure will need at a minimum the 5 operations discussed during the lectures, as shown below. You can rename these operations away from the naming conventions of the MIT textbook, as long as it is clear what each of them does.

- BUILD-MAX-HEAP: Takes an arbitrary array and builds it into a max heap
- MAX-HEAPIFY: Takes an almost-heap with one violation, and fixes the violation
- HEAP-MAXIMUM: Returns the largest element in the max heap
- HEAP-EXTRACT-MAX: Removes and returns the largest element in the max heap
- MAX-HEAP-INSERT: Inserts a new element into the heap, preserving the heap property

In addition to these operations, you should also implement two display methods, for testing purposes.

- printAsArray: Prints the array representation (e.g. [16,14,10,8,7,3,9,1,4,2])
- printAsTree: Prints the heap as a sideways tree, as shown below:

```
      9
   10
      3
16
      7
         2
   14
         4
      8
         1
```

Hint: For printAsTree, you'll find it easier to write the function recursively including a depth argument (which indicates how far to the right to indent the output). Recursion will be used to print the left and right subtrees (with a incremented depth).

The implementation of hashsort() will be a function that takes an arbitrary array, and sorts it using a heap. Be sure to include testing code, where you create a heap from an arbitrary array, use buildheap, and then heapsort the array (printing the array before and after), as well as testing the other operations mentioned above.

Part 2

Read the first 20 or so slides of the document from Stanford NLP group on Minimum Edit Distance (MED):

http://www.stanford.edu/class/cs124/lec/med.pdf

Using this technique, implement this algorithm using Dynamic Programming in an approved programming language. Include some testing code to try it out on a few string pairs:

- spoof/stool
- podiatrist/pediatrician
- blaming/conning

How to Submit

Put your source code answers to part 1 and 2 into a ZIP file, called Assignment2_FirstNameLastName_StudentNum.zip (do not use RAR or other archival formats) and submit this file to Blackboard.