CSC 225 SUMMER 2014
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 3 - PROGRAMMING
UNIVERSITY OF VICTORIA

1 Programming Assignment

The programming assignment is to implement the heap sort algorithm with the following input and output specification:

Input: An array A of n integers.
Output: The elements of A will be written back to A in sorted order.

Pseudocode for heap sort is given below:

```
procedure HeapSort($A, n$)
    $H \leftarrow$ Empty Heap
    for $i = 0, 1, \ldots, n-1$ do
        $H$.Insert($A[i]$)
    end for
    for $i = 0, 1, \ldots, n-1$ do
        $A[i] \leftarrow H$.RemoveMin( )
    end for
end procedure
```

A Java template has been provided containing an empty function `HeapSort`, which takes an integer array A as its only argument. Your task is to write the body of the `HeapSort` function. You must use the provided Java template as the basis of your submission, and put your implmentation inside the `HeapSort` function in the template. You may not change the name, return type or parameters of the `HeapSort` function. You may add additional functions as needed. The `main` function in the template contains code to help you test your implementation by entering test data or reading it from a file. You may modify the `main` function, but only the contents of the `HeapSort` function (and any functions you have added) will be marked. Please read through the comments in the template file before starting.

2 Test Datasets

The test arrays uploaded for assignments 1 and 2 can be used to test your heap sort implementation. A small set of additional files has also been uploaded.

The uploaded files may not cover all possible cases, so you should test your implementation on other inputs as well.

3 Evaluation Criteria

The programming assignment will be marked out of 25, based on a combination of automated testing (using large test arrays similar to the ones posted on conneX) and human inspection.

Score (/25)	**Description**
$0 - 3$	Submission does not compile or does not conform to the provided template.
$3 - 15$	The implemented algorithm is not heap sort or is substantially inaccurate on the tested inputs.
$16 - 20$	The implemented algorithm is heap sort but does not have a $\Theta(n \log n)$ running time or is inaccurate on some inputs.
$21 - 25$	The implemented algorithm is correct and has a $\Theta(n \log n)$ running time.

To be properly tested, every submission must compile correctly as submitted, and must be based on the provided template. **If your submission does not compile for any reason (even trivial mistakes like typos), or was not based on the template, it will receive at most 3 out of 25**. The best way to make sure your submission is correct is to download it from conneX after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. conneX will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, conneX will automatically send you a confirmation email. If you do not receive such an email, your submission was not received. If you have problems with the submission process, send an email to the instructor **before** the due date.