CSC 225 - SPRING 2018
ALGORITHMS AND DATA STRUCTURES I
PROGRAMMING ASSIGNMENT 2
UNIVERSITY OF VICTORIA

1 Programming Assignment

Your job is to design a data structure that stores integers and supports insert in logarithmic time, find the median in constant time, and delete the median in logarithmic time. To achieve these guarantees, your data structure will require the use of two heaps. As a hint, think about how two heaps could store the contents of the data structure while also being beneficial for the insert, find median, and delete median operations.

A Java template has been provided containing three classes: `median`, `minHeap` and `maxHeap`. You will fill out the body of the functions that are not implemented. Your main task is to write the body of the `CalculateMedian` function, which will use the the functions in the `minHeap` and `maxHeap` classes you implement. You may assume that the input will contain no negative values and there will always be an odd number of integers. Your code is not required to check for incorrectly formed input data.

You must use the provided Java template as the basis of your submission. You may not change the name, return type or parameters of any of the functions, but may add additional functions if you need. The main function in the template contains code to help you test your implementation by entering test data manually. You may modify the main function, since the main function will be deleted before marking begins. You can submit ONLY the file named "median.java". If you submit multiple files or change the name of the class or the functions (even lowercase to uppercase or vice versa), your submission may generate compilation error during the automated testing and you will be graded accordingly.

2 Examples

The samples below show the correct output for various test inputs.

Sample input output 1
Enter a list of non negative integers. To end enter a negative integer.
45
current median:45
65
55
current median:55
1

0
current median:45
52
98
current median:52
65
32
current median:52
54
23
current median:52
-9

Sample input output 2
Enter a list of non negative integers. To end enter a negative integers.
4
current median:4
2
6
current median:4
0
0
current median:2
0
0
current median:0
9
5
current median:2
6
8
current median:4
8
8
current median:5
8
8
current median:6
8
8
current median:6
8
8
current median:8
-1

Sample input output 3
Enter a list of non negative integers. To end enter a negative integers.
5
current median:5
6
9
current median:6
99
8569
current median:9
45
12
current median:12
9999
9877
current median:45
65893
98456
current median:99
895623
986565
current median:8569
-1

3 Evaluation Criteria

The programming assignment will be marked out of 50, based on a combination of automated testing and human inspection. You may safely assume that the maximum size of an input array will be 10000.

The mark for each submission will be based on both the asymptotic worst case running time and the ability of the algorithm to handle inputs of different sizes. The table below shows the expectations associated with different scores.

Score	**Description**
0 - 10	Submission does not compile or does not conform to the provided template.
11 - 25	The submission is substantially inaccurate on the tested inputs (fails more than 50% of the test cases).
26 - 30	The implemented data structure takes more than $O(\log n)$ time to insert or delete the median or more than $O(1)$ time to find the median.
31 - 50	The implemented data structure takes $O(\log n)$ time to insert or delete the median and $O(1)$ time to find the median. The numbers would vary based on how many test cases your algorithm gives correct answer to.

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 10 out of 50. 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.