University of Waterloo
CS240 - Spring 2023
Programming Question 1

Due Date: Wednesday June 14, 5pm

You should have submitted AID01 before the due date of this assignment. The agreement will indicate what you must do to ensure the integrity of your grade. If you are having difficulties with the assignment, course staff are there to help (provided it isn't last minute).

The Academic Integrity Declaration must be signed and submitted on time or the assessment will not be marked.

Please read `https://student.cs.uwaterloo.ca/~cs240/s23/assignments.phtml#guidelines` for guidelines on submission. Submit the file `report.cpp` to ~~Markus~~ Marmoset.

Late Policy: Assignments are due at 5:00pm, with the grace period until 11:59pm. Assignments submitted after 11:59pm on the due date will not be accepted but may be reviewed (by request) for feedback purposes only.

Consider a max-heap H that holds n distinct integers, implemented as an array (you can access all entries in the array). We want to report the k largest integers from H, in decreasing order, without modifying H; we assume $k \leq n$.

For that, we maintain an auxiliary heap B that will contain $O(k)$ pairs of integers $(i, H[i])$. The first entry in such a pair is the index of an element in H, and the second entry is the integer stored at that index in H. Comparisons in B are done using the second entry $H[i]$. Initially, we insert $(0, H[0])$ in B.

Using heap B as above, design and implement an algorithm that reports the k largest integers from H in time $O(k \log(k))$. Your algorithm should not modify H, but can modify B using `insert` or `deleteMax`.

We give you a skeleton that you should complete. **Please do not modify anything up to, and including, the main function**, as doing so would prevent our automated tests from running correctly (which would affect your grade). The skeleton provides a templated heap class (which you have to complete), and declares a function `report` that you have to write.

The main function reads from `cin` the size n, then the n integers in the heap $H[0 \ldots n-1]$, and finally the integer k, and calls the `report` function. This function will have to write to `cout` the k largest integers in H, one per line, in decreasing order. We print the heap H

before and after calling `report` (to make sure it did not change), and time *only* the reporting function. So for instance if the input consists of the following lines:

```
5
17
15
10
3
13
3
```

then you should see something like this as output:

```
17 15 10 3 13
17
15
13
17 15 10 3 13
1.3556e-05
```

The first line is the heap H, then we see its $k = 3$ largest entries in decreasing order, then we see H again, and finally the time for `report` (this will of course vary). You should not use STL `priority_queue`'s. You can use `int`'s to store all your integers. After completing the skeleton we give you, rename it and submit it as a file called `report.cpp`.