University of Waterloo
CS240E, Spring 2025
Programming Assignment 1

Due Date: Tuesday, June 10, 2023 at 5pm

Be sure to read the assignment guidelines (`https://student.cs.uwaterloo.ca/~cs240e/s25/assignments.phtml#guidelines`). Submit your solution to Marmoset. Ensure you have read, signed, and submitted the Academic Integrity Declaration AID01.

Grace period: submissions made before 8:00pm on June 10 will be accepted without penalty. Please note that submissions made after 8pm will not be graded and may only be reviewed for feedback.

Question 1 [20 marks]

Implement a variant of a priority queue that supports both `deleteMin` and `deleteMax` in $O(\log n)$ time, where n is the number of items stored. Thus, your class `MinMaxPQ` must have the following methods:

- `void insert(int p)`: inserts an item with key p
- `int deleteMin()`: returns the key of the item that had the minimum key of all stored items, and *also removes that item*
- `int deleteMax()`: returns the key of the item that had the maximum key of all stored items. It also removes that item.

Note that keys may occur repeatedly. To support these operations, you should implement a min-max-heap. This is a tree that has the structural property of a binary heap (and hence can be stored in an array), but the order-property is different:

- for any node x on level $2, 4, 6, \ldots$: $key(parent(parent(x))) \leq key(x) \leq key(parent(x))$
- for any node x on level $3, 5, \ldots$: $key(parent(x)) \leq key(x) \leq key(parent(parent(x)))$
- for any node x on level 1 (i.e. a child of the root): $key(parent(x)) \leq key(x)$

Figuring out the details of how to implement the operations is part of your assignment. All of `insert`, `deleteMin`, `deleteMax` must take $O(\log n)$ time, where n is the current size of the array.

For testing purposes, your class must additionally support the following operations, all of which should take constant time:

- constructor and destructor for `MinMaxPQ`

- `int size()`: returns the current size
- `int priorityAt(int i)`: returns the key of the item that would be stored at $A[i]$ if the *heap were stored in an array in level-order*. For example, `priorityAt(0)` returns the key at the root.

Submit a file `MinMaxPQ.cpp` which contains the source code of all your classes. The provided file gives a stub. File `MinMaxPQ.cpp` should not have a main-routine, or it should be surrounded by an `#ifndef TESTING` as done in the stub.

Submit your solution to Marmoset. Marmoset will be set up to translate your program with `g++ -std=c++17`.

For this assignment, you *are* allowed to use `vector` from the STL. You must not have memory leaks.