

University of Waterloo

CS240 - Spring 2024

Programming Question 1

Due Date: Tuesday June 11, 5pm

Please read the following link for guidelines on submission:

<https://student.cs.uwaterloo.ca/~cs240/s24/assignments.phtml#guidelines>

Submit the file `two_sided_pq.cpp` to Marmoset.

Late Policy: Assignments are due at 5:00pm, with the grace period until 11:59pm.

You are going to implement a *double-ended priority queue* of integers. This is a collection where we can insert integers, access the minimum or maximum and remove the minimum or maximum; our goal is do to `insert`, `delete_min`, `delete_max` in time $O(\log(n))$ (if there are n elements in the collection) and `get_min`, `get_max` in constant time.

To support these requirements, you will use two heaps, called `min_heap` (a min-heap) and `max_heap` (a max-heap) stored as STL vectors (the assignment guidelines say you should in general use arrays, not vectors, but here it will be convenient not to have to implement dynamic arrays yourselves). At all times, these heaps should contain the same set of elements, but stored in a different order. You will also use two arrays `indices_of_min_in_max` and `indices_of_max_in_min` that specify the correspondence between the heaps: at all times, `indices_of_min_in_max[i]` should give the index of `min_heap[i]` in `max_heap`, and `indices_of_max_in_min[i]` should give the index of `max_heap[i]` in `min_heap`.

Design and implement the algorithms needed to support the operations listed above, with the required worst-case runtimes. We give you a skeleton that you should complete, where we declared a class `two_sided_PQ` with a few helper functions. The functions you have to complete are the constructor, `get_size`, `get_min`, `get_max`, `delete_min`, `delete_max`, `insert`. **Please do not modify anything in the body of `two_sided_PQ` itself, or the `main` function**, as doing so would prevent our automated tests from running correctly (which would probably affect your grade).

The `main` function creates a `two_sided_PQ` T then reads its input from `cin`. The input is a sequence of instructions (one per line) of the following kind: `insert v` (v is an integer), `delete min`, `delete max`, `print all`. For each instruction, `main` calls the corresponding function in `two_sided_PQ` and prints the new min and max in T (except if the instruction is `print`, in which case we print everything using the given `print` function in `two_sided_PQ`). For instance, with this input

```
insert 1
insert 2
print all
delete max
```

then you should see something like this as output:

```
after insert 1: 1 1
after insert 2: 1 2
print all:
size: 2
max heap: 2 1
min heap: 1 2
indices of max in min: 1 0
indices of min in max: 1 0
after delete max: 1 1
```

Note: our automatic tests will not use the print all instruction; we provide it in order to help with debugging.

You should not use STL `priority_queue`'s. After completing the skeleton we give you, submit it as a file called `two_sided_pq.cpp`.