

## lab 20 Priority Queues

---

**Instructions:** In this lab implement a priority queue using something better than  $O(n)$  for add and remove.

Implement the following interface:

```
1 #ifndef PRIORITY_QUEUE_H
2 #define PRIORITY_QUEUE_H
3
4 template<class T>
5 class PriorityQueue {
6     private:
7         /* Class to implement.*/
8     public:
9         /* Empty constructor shall create an empty PriorityQueue! */
10        PriorityQueue();
11
12        /* Do a deep copy of queue into the this.
13         * Note: This one uses a reference to a PriorityQueue!
14         */
15        PriorityQueue(const PriorityQueue<T> &pq);
16
17        /* Deconstructor shall free up memory */
18        ~PriorityQueue();
19
20        /* Return the current length (number of items) in the queue */
21        int getLength() const;
22
23        /* Returns true if the queue is empty. */
24        bool isEmpty() const;
25
26        /* Print out the PriorityQueue */
27        void print() const;
28
29        /* Pushes the val to the top of the queue. */
30        bool push(const T &val);
31
32        /* Removes and returns the top (largest) element from the queue. */
33        T pop();
34
35        /* Returns if the two lists contain the same elements in the
36         * same order.
37         */
38        bool operator==(const PriorityQueue<T> &pq) const;
39 };
40
```

```
41 #include "priorityqueue.cpp"
42
43 #endif
```

**Write some test cases:**

Create some test cases, using Unity, that you believe would cover all aspects of your code.

**STL:**

You may use the STL, but not queue or priority\_queue.

**Memory Management:**

Now that are using new, we must ensure that there is a corresponding delete to free the memory. Ensure there are no memory leaks in your code! Please run Valgrind on your tests to ensure no memory leaks!

**How to turn in:**

Turn in via GitHub. Ensure the file(s) are in your directory and then:

- \$ git add <files>
- \$ git commit
- \$ git push

**Due Date:** November 17, 2021 2359

**Teamwork:** No teamwork, your work must be your own.