```
1 // COS30008, Final Exam, 2024
 3 #pragma once
 4
 5 #include <optional>
 6 #include <cassert>
 7
 8 #include <iostream>
 9
10 template<typename T>
11 class DynamicQueue
12 {
13 private:
14
       T* fElements;
15
        size_t fFirstIndex;
        size_t fLastIndex;
16
17
       size_t fCurrentSize;
18
19
       void resize(size_t aNewSize) {
20
            T* lNewElements = new T[aNewSize];
21
            size_t j = 0;
22
            for (size_t i = fFirstIndex; i < fLastIndex; ++i, ++j) {</pre>
23
                lNewElements[j] = std::move(fElements[i]);
24
            }
25
            delete[] fElements;
            fElements = lNewElements;
26
27
            fFirstIndex = 0;
28
            fLastIndex = j;
29
            fCurrentSize = aNewSize;
30
        }
31
32
        void ensure_capacity() {
33
            if (fLastIndex >= fCurrentSize) {
34
                resize(fCurrentSize * 2);
35
            }
        }
36
37
38
        void adjust_capacity() {
            if ((fLastIndex - fFirstIndex) <= fCurrentSize / 4 &&</pre>
39
              fCurrentSize > 1) {
40
                resize(fCurrentSize / 2);
            }
41
42
        }
43
44
45
   public:
        DynamicQueue() : fElements(new T[1]), fFirstIndex(0), fLastIndex(0), >
46
           fCurrentSize(1) {}
47
        ~DynamicQueue() {
48
49
            delete[] fElements;
        }
50
51
```

```
D:\COS30008\Programs\Final\DynamicQueue.h
```

```
•
```

```
52
       DynamicQueue(const DynamicQueue&) = delete;
       DynamicQueue& operator=(const DynamicQueue&) = delete;
53
54
       std::optional<T> top() const noexcept {
55
            if (fFirstIndex == fLastIndex) {
56
57
                return std::nullopt;
            }
58
            return fElements[fFirstIndex];
59
       }
60
61
       void enqueue(const T& aValue) {
62
63
            ensure_capacity();
            fElements[fLastIndex++] = aValue;
64
65
       }
66
67
       void dequeue() {
68
            if (fFirstIndex < fLastIndex) {</pre>
69
                ++fFirstIndex;
70
                adjust_capacity();
            }
71
72
       }
73
74 };
75
```