Chapter 7

Stacks and Queues

Stacks and queues are linear data structures that store lists of values. They differ in the operations that store and retrieve the data.

Stacks store data with the `push()` operation and retrieve data with the `pop()` operation. The order is last-in first-out, so stacks are known as LIFO lists. Queues store data with the `enqueue()` operation and retrieve data with the `dequeue()` operation. The order is first-in first-out, so queues are known as FIFO lists.

7.1 Array Implementations

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The array implementation of a stack

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
// ========= StackA =========
template<class T>
class StackA {
private:
  ArrayT<T> _data;
  int _top;

public:
  StackA(int cap);
  // Post: This stack is allocated with a capacity of cap
  // and initialized to be empty.
  bool isEmpty() const;
  // Post: true is returned if this stack is empty;
  // otherwise, false is returned.
  bool isFull() const;
  // Post: true is returned if this stack is full;
  // otherwise, false is returned.
  T pop();
  // Pre: This stack is not empty.
  // Post: The top value in this stack is removed and returned.
  void push(T const &val);
  // Pre: This stack is not full.
  // Post: val is stored on top of this stack.
  T const &topOf() const;
  // Pre: This stack is not empty.
  // Post: The top value from this stack is returned.
  void toStream(ostream &os) const;
  // Post: All the items on this stack from top to bottom
  // are written to os.
};

Figure 7.1 StackA.hpp. Specification for the StackA data structure. The listing continues in Figure 7.3

about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you
7.1 Array Implementations

Figure 7.2 Action of `push()` and `pop()` with the StackA data structure.

will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
// ======== Constructor ========
template<class T>
StackA<T>::StackA(int cap):
    _data(cap),
    _top(-1) {
}

// ======== isEmpty ========
template<class T>
bool StackA<T>::isEmpty() const {
    cerr << "isEmpty: Exercise for the student."
         << endl;
    throw -1;
}

// ======== isFull ========
template<class T>
bool StackA<T>::isFull() const {
    cerr << "isFull: Exercise for the student."
         << endl;
    throw -1;
}

// ======== pop ========
template<class T>
T StackA<T>::pop() {
    if (isEmpty()) {
        cerr << "pop precondition violated: "
             << "Cannot pop from an empty stack."
             << endl;
        throw -1;
    }
    cerr << "pop: Exercise for the student."
         << endl;
    throw -1;
}

// ======== push ========
template<class T>
void StackA<T>::push(T const &val) {
    cerr << "push: Exercise for the student."
         << endl;
    throw -1;
}

**Figure 7.3** StackA.hpp (continued). Operations for the StackA data structure. The listing continues in the next figure.
### 7.1 Array Implementations

```cpp
// ======== topOf ========
template<class T>
T const &StackA<T>::topOf() const {
    if (isEmpty()) {
        cerr << "topOf precondition violated: " << endl;
        throw -1;
    }
    cerr << "topOf: Exercise for the student." << endl;
    throw -1;
}

// ======== operator<< ========
template<class T>
ostream &operator<<(ostream &os, StackA<T> const &rhs) {
    rhs.toStream(os);
    return os;
}

// ======== toStream ========
template<class T>
void StackA<T>::toStream(ostream &os) const {
    os << "(";
    for (int i = _top; i > 0; i--) {
        os << _data[i] << ", ";
    }
    if (_top == -1) {
        os << ");"
    } else {
        os << _data[0] << ");"
    }
}
```

**Figure 7.4** StackA.hpp (continued). Output for the StackA data structure. This concludes the listing.

There is no need for special content, but the length of words should match the language.
Chapter 7  Stacks and Queues

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.

The circular queue

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.
// ========= QueueA =========
template<class T>
class QueueA {
private:
    ArrayT<T> _data;
    int _head, _tail;

public:
    QueueA(int cap);
    // Post: This queue is allocated with a capacity of cap
    // and initialized to be empty.

    T dequeue();
    // Pre: This queue is not empty.
    // Post: The head value in this queue is removed and returned.

    void enqueue(T const &val);
    // Pre: This queue is not full.
    // Post: val is stored at the tail of this queue.

    T const &headOf() const;
    // Pre: This queue is not empty.
    // Post: The head value from this queue is returned.

    bool isEmpty() const;
    // Post: true is returned if this queue is empty;
    // otherwise, false is returned.

    bool isFull() const;
    // Post: true is returned if this queue is full;
    // otherwise, false is returned.

    void toStream(ostream &os) const;
    // Post: All the items on this queue from tail to head
    // are written to os.
};

Figure 7.5 QueueA.hpp. Specification for the QueueA data structure. The listing continues in Figure 7.7
(a) Initial state.

(b) enqueue(3.3);

(c) enqueue(5.5);

(d) enqueue(4.4);

(e) dequeue();
Returns 3.3.

(f) enqueue(9.9);

(g) enqueue(1.1);

(h) dequeue();
Returns 5.5.

(i) dequeue();
Returns 4.4.

(j) enqueue(7.7);

(k) enqueue(6.6);

(l) enqueue(2.2);
The queue is full.

Figure 7.6 Action of enqueue() and dequeue() with the QueueA data structure.
7.2 The Adapter Pattern

The adapter pattern is also known as a wrapper.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Figure 7.7 QueueA.hpp (continued). Constructor and output for the QueueA data structure. Operations that are exercises for the student are not shown. This concludes the listing.

```cpp
// ======== Constructor ========
template<class T>
QueueA<T>::QueueA(int cap) :
    _data(cap + 1),
    _head(0),
    _tail(0) {
}

// ======== operator<< ========
template<class T>
ostream &operator<<(ostream &os, QueueA<T> const &rhs) {
    rhs.toStream(os);
    return os;
}

// ======== toStream ========
template<class T>
void QueueA<T>::toStream(ostream &os) const {
    cerr << "toStream: Exercise for the student." << endl;
    throw -1;
}
```

language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

7.2 The Adapter Pattern

The adapter pattern is also known as a wrapper.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original

Revised: October 8, 2016
Copyright ©: 1998, Dung X. Nguyen and J. Stanley Warford
// ======== StackL ========
template<class T>
class StackL {
private:
    ListL<T>* _listL;

public:
    StackL();  // This stack is initialized to be empty.
    ~StackL(); // Post: This stack is deallocated.
    bool isEmpty() const; // Post: true is returned if this stack is empty;
                         // otherwise, false is returned.
    T pop(); // Pre: This stack is not empty.
             // Post: The top value in this stack is removed and returned.
    void push(T const &val); // Post: val is stored on top of this stack.
    T const &topOf() const; // Pre: This stack is not empty.
                            // Post: The top value from this stack is returned.
    void toStream(ostream &os) const; // Post: All the items on this stack from top to bottom
                                       // are written to os.
};

Figure 7.8  StackL.hpp. Specification for the StackL data structure. The listing continues in Figure 7.10.

language. There is no need for special content, but the length of words should match the language.

The list implementation of a stack

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This
7.2 The Adapter Pattern

Figure 7.9 The adapter design pattern for the `StackL` data structure.

text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the
// ======== Constructor ========
template<class T> 
StackL<T>::StackL() {
    _listL = new ListL<T>();
}

// ======== Destructor ========
template<class T>
StackL<T>::~StackL() {
    delete _listL;
}

// ======== isEmpty ========
template<class T>
bool StackL<T>::isEmpty() const {
    return _listL->isEmpty();
}

// ======== push ========
template<class T>
void StackL<T>::push(const T &val) {
    _listL->prepend(val);
}

// ======== operator<< ========
template<class T>
ostream &operator<<(ostream &os, StackL<T> const &rhs) {
    rhs.toStream(os);
    return os;
}

// ======== toStream ========
template<class T>
void StackL<T>::toStream(ostream &os) const {
    _listL->toStream(os);
}

**Figure 7.10** StackL.hpp (continued). Operations for the StackL data structure. Operations that are exercises for the student are not shown. This concludes the listing.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This
7.2 The Adapter Pattern

```cpp
// ========= QueueL =========
template<class T>
class QueueL {
    private:
        // Attribute is exercise for the student.

    public:
        QueueL();
        // This queue is allocated and initialized to be empty.

        ~QueueL();
        // Post: This queue is deallocated.

        T dequeue();
        // Pre: This queue is not empty.
        // Post: The head value in this queue is removed and returned.

        void enqueue(T const &val);
        // Post: val is stored at the tail of this queue.

        T const &headOf() const;
        // Pre: This queue is not empty.
        // Post: The head value from this queue is returned.

        bool isEmpty() const;
        // Post: true is returned if this queue is empty;
        // otherwise, false is returned.

        void toStream(ostream &os) const;
        // Post: All the items on this queue from tail to head
        // are written to os.
};
```

**Figure 7.11** QueueL.hpp. Specification for the QueueL data structure. The implementation is an exercise for the student and is not shown.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

The list implementation of a queue

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
about the selected font, how the letters are written and an impression of the look. This
text should contain all letters of the alphabet and it should be written in of the original
language. There is no need for special content, but the length of words should match the
language.

7.3 The Priority Queue

Hello, here is some text without a meaning. This text should show what a printed text
will look like at this place. If you read this text, you will get no information. Really?
Is there no information? Is there a difference between this text and some nonsense like
“Huardest gefburn”? Kjift – not at all! A blind text like this gives you information
about the selected font, how the letters are written and an impression of the look. This
text should contain all letters of the alphabet and it should be written in of the original
language. There is no need for special content, but the length of words should match the
language.

The heap implementation of a priority queue

This is the second paragraph. Hello, here is some text without a meaning. This text
should show what a printed text will look like at this place. If you read this text, you
will get no information. Really? Is there no information? Is there a difference between
this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text
like this gives you information about the selected font, how the letters are written and an
impression of the look. This text should contain all letters of the alphabet and it should
be written in of the original language. There is no need for special content, but the length
of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text
without a meaning. This text should show what a printed text will look like at this place.
If you read this text, you will get no information. Really? Is there no information? Is
there a difference between this text and some nonsense like “Huardest gefburn”? Kjift
– not at all! A blind text like this gives you information about the selected font, how the
letters are written and an impression of the look. This text should contain all letters of
the alphabet and it should be written in of the original language. There is no need for
special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some
text without a meaning. This text should show what a printed text will look like at this
place. If you read this text, you will get no information. Really? Is there no information?
Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift
– not at all! A blind text like this gives you information about the selected font, how the
letters are written and an impression of the look. This text should contain all letters of
the alphabet and it should be written in of the original language. There is no need for
special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed
text will look like at this place. If you read this text, you will get no information. Really?
Is there no information? Is there a difference between this text and some nonsense like
“Huardest gefburn”? Kjift – not at all! A blind text like this gives you information
about the selected font, how the letters are written and an impression of the look. This
template<class T>
class PriorityQ {
private:
    ArrayT<T> _data;
    int _hiIndex;

public:
    PriorityQ(int cap);
    // Post: This priority queue is allocated with a capacity of cap
    // and initialized to be empty.

    T extractMax();
    // Pre: This priority queue is not empty.
    // Post: The maximum value in this priority queue is removed
    // and returned.

    int heapSize() const;
    // Post: The size of this priority queue is returned.

    void increaseKey(int i, T const &key);
    // Pre: This priority queue is not empty, 0 <= i < heapSize(),
    // and key is at least as large as the key at index i.
    // Post: The value of the element at index i is increased to key.

    void insert(T const &val);
    // Pre: This priority queue is not full.
    // Post: val is stored in this priority queue.

    bool isEmpty() const;
    // Post: true is returned if this priority queue is full;
    // otherwise, false is returned.

    bool isFull() const;
    // Post: true is returned if this priority queue is full;
    // otherwise, false is returned.

    T const &maximum() const;
    // Pre: This priority queue is not empty.
    // Post: The maximum value from this priority queue is returned.

    void toStream(ostream &os) const;
    // Post: Each item on this priority queue prefixed with its index
    // is written to os.
};
text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.
text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like
7.3 The Priority Queue

```cpp
// ========= siftUp ========
template<class T>
void siftUp(ASeq<T> &a, int lo, int i) {
    // Pre: maxHeap(a[lo..i - 1]).
    // Post: maxHeap(a[lo..i]).
    T temp = a[i];
    int parent = (i + lo - 1) / 2;
    while (lo < i && a[parent] < temp) {
        cerr << "siftUp: Exercise for the student." << endl;
        throw -1;
    }
    a[i] = temp;
}
```

Figure 7.14 Heapifier.hpp in the ASorter project. Implementation of the siftUp() function.

“Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Exercises

7–1 What is the worst case asymptotic bound $\Theta(n)$, where $n$ is the number of elements stored in the data structure, for each of the following operations?

(a) push() for the StackA data structure.
(b) pop() for the StackA data structure.
// ========= Constructor =========
template<class T>
PriorityQ<T>::PriorityQ(int cap) :
    _data(cap),
    _hiIndex(-1) {
}

// ========= heapSize =========
template<class T>
int PriorityQ<T>::heapSize() const {
    return _hiIndex + 1;
}

// ========= insert =========
template<class T>
void PriorityQ<T>::insert(T const &val) {
    if (isFull()) {
        cerr << "insert precondition violated: "
        << "Cannot insert into a full priority queue." << endl;
        throw -1;
    }
    _data[++_hiIndex] = val;
    siftUp(_data, 0, _hiIndex);
}

// ========= isEmpty =========
template<class T>
bool PriorityQ<T>::isEmpty() const {
    return _hiIndex == -1;
}

// ========= maximum =========
template<class T>
T const &PriorityQ<T>::maximum() const {
    if (isEmpty()) {
        cerr << "maximum precondition violated: "
        << "An empty priority queue has no maximum." << endl;
        throw -1;
    }
    return _data[0];
}

Figure 7.15  PriorityQ.hpp (continued). Operations for the PriorityQ data structure. Operations that are exercises for the student are not shown. The listing continues in the next figure.
7.3 The Priority Queue

// ======== operator<< ========
template<class T>
ostream &operator<<(ostream &os, PriorityQ<T> const &rhs) {
    rhs.toStream(os);
    return os;
}

// ======== toStream ========
template<class T>
void PriorityQ<T>::toStream(ostream &os) const {
    for (int i = 0; i <= _hiIndex; i++) {
        os << i << ":" << _data[i] << " ";
    }
}

Figure 7.16 PriorityQ.hpp (continued). The output operations for the PriorityQ data structure. This concludes the listing.

(c) enqueue() for the QueueA data structure.
(d) dequeue() for the QueueA data structure.
(e) push() for the StackL data structure.
(f) pop() for the StackL data structure.
(g) enqueue() for the StackL data structure.
(h) dequeue() for the StackL data structure.
(i) insert() for the PriorityQ data structure.
(j) extractMax() for the PriorityQ data structure.
(k) maximum() for the PriorityQ data structure.
(l) increaseKey() for the PriorityQ data structure.

7–2 Implement the methods isEmpty(), isFull(), push(), pop(), and topOf() for the StackA data structure.

7–3 Implement the methods isEmpty(), isFull(), enqueue(), dequeue(), headOf(), and toStream() for the QueueA data structure.

7–4 Implement the methods pop() and topOf() for the StackL data structure.

7–5 Implement the attribute, the constructor, the destructor, and the methods isEmpty(), enqueue(), dequeue(), headOf(), operator<<(), and toStream() for the QueueL data structure.

7–6 Implement the methods isFull(), siftUp(), extractMax(), and increaseKey() for the PriorityQ data structure.
(a) \texttt{insert}(80);

(b) \texttt{data}[++hiIndex] = val;

(c) \texttt{siftUp}(_data, 0, _hiIndex);

\textbf{Figure 7.17} Action of \texttt{insert()} with the \texttt{PriorityQ} data structure.