Renteria Magana Rayni Damian

Double-Ended Queues

We next consider a queue-like data structure that supports insertion and deletion at both the front and the back of the queue. Such a structure is called a double-ended queue, or deque, which is usually pronounced “deck” to avoid confusion with the dequeue method of the regular queue ADT, which is pronounced like the abbreviation “D.Q.”

The deque abstract data type is more general than both the stack and the queue ADTs. The extra generality can be useful in some applications. For example, we described a restaurant using a queue to maintain a waitlist. Occassionally, the first person might be removed from the queue only to find that a table was not available; typically, the restaurant will re-insert the person at the first position in the queue. It may also be that a customer at the end of the queue may grow impatient and leave the restaurant. (We will need an even more general data structure if we want to model customers leaving the queue from other positions.

To provide a symmetrical abstraction, the deque ADT is defined so that deque D **supports the following methods**:

D.add first(e): Add element e to the front of deque D.

D.add last(e): Add element e to the back of deque D.

D.delete first( ): Remove and return the first element from deque D;
an error occurs if the deque is empty.

D.delete last( ): Remove and return the last element from deque D;
an error occurs if the deque is empty.

Additionally, the deque ADT will include the **following accessors:**

D.first( ): Return (but do not remove) the first element of deque D;
an error occurs if the deque is empty.

D.last( ): Return (but do not remove) the last element of deque D;
an error occurs if the deque is empty.

D.is empty( ): Return True if deque D does not contain any elements.

len(D): Return the number of elements in deque D; in Python,
we implement this with the special method len .

References

Goodrich, M. T., Goldwasser, M. H., & Tamassia, R. (2013). *Data structures and algorithms in Python*. Wiley.