ECE 2574: Data Structures and Algorithms - Deque Implementation

C. L. Wyatt

Today we will look at an improved Deque implementation using pointers to blocks.

In a couple of weeks we will see a better implementation of priority queues using Heaps.

- Review Deque ADT
- Review: array-based and link-based deque implementations
- Deque implementation using linked blocks

Review: Double-ended Queue (deque) ADT

A Queue in which you can enqueue or dequeue at either end is called a double-ended queue or *deque* (pronounced "deck").

```
+isEmpty(): boolean
+enqueue_front(newEntry: ItemType): boolean
+dequeue_front(): boolean
+peekFront(): ItemType
+enqueue_back(newEntry: ItemType): boolean
+dequeue_back): boolean
+peekBack(): ItemType
```

This gives a combination of a stack and a queue. See `abstract_deque.h`.

Performance of array-based and link-based (double) deque implementations

Working with a partner fill in the following table of time complexity and space complexity for both implementations.

operation	Array		Link	
	Best	Worst	Best	Worst
enqueue_front				
dequeue_front				
peekFront				
enqueue_back				
dequeue_back				
peekBack				

Performance of array-based and link-based (double) deque implementations

Time complexity:

operation	Array		Link	
	Best	Worst	Best	Worst
enqueue_front	O(1)	O(n)	O(1)	
dequeue_front	O(1)		O(1)	
peekFront	O(1)		O(1)	
enqueue_back	O(1)	O(n)	O(1)	
dequeue_back	O(1)		O(1)	
peekBack	O(1)		O(1)	

- In practice the constants associated with the link-based operations can make performance worse than it could be.

Deque implementation using arrays of arrays

There are several variations and different names for this:
e.g. chuncklists, vlists
See `deque.h` and `deque.txx`

Next Actions and Reminders

- Read CH pp. 415-421 on operator overloading
- Program 4 (parts I and II) is due 11/17