

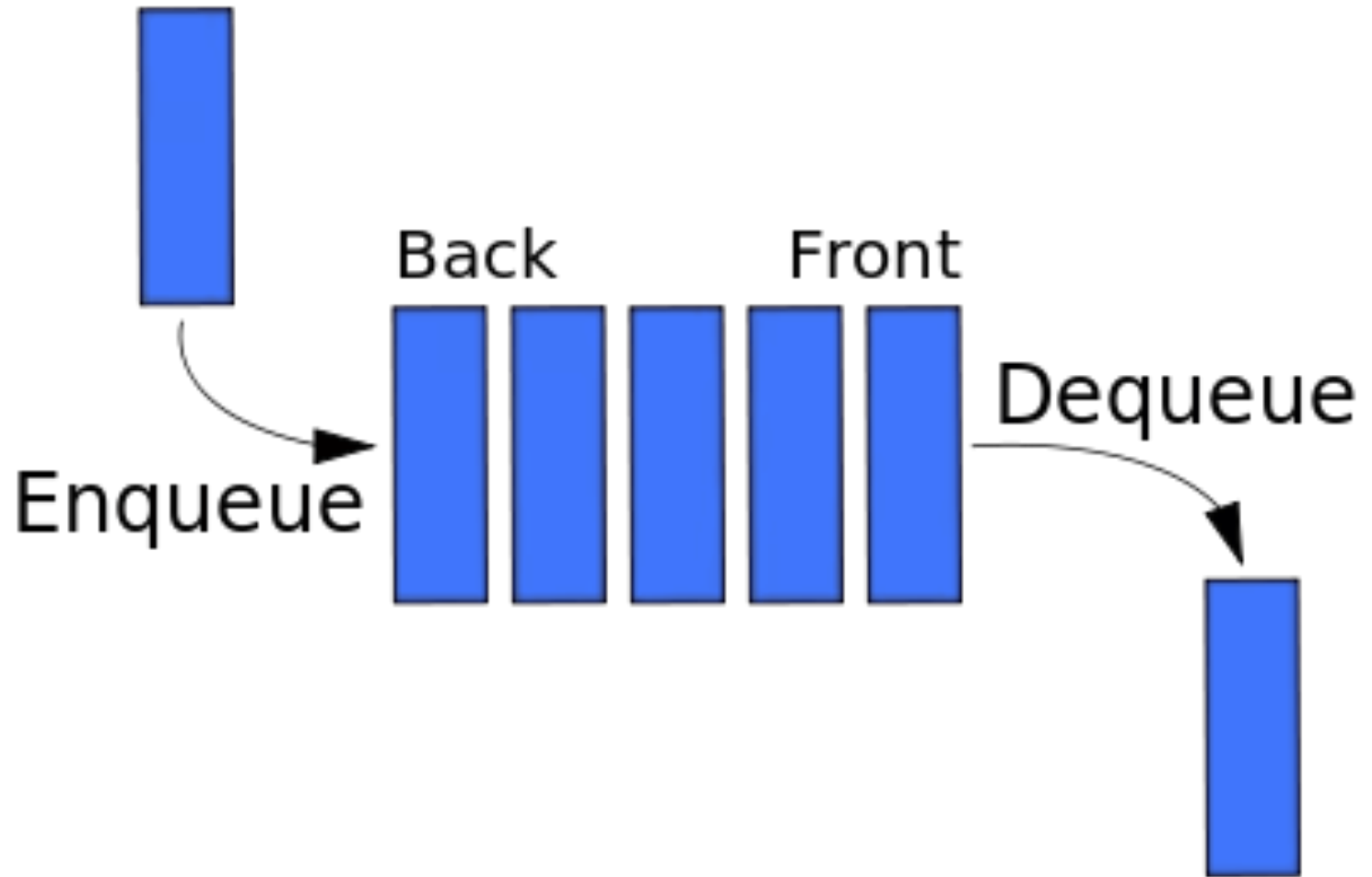
Queues

CSCI-2270

Elizabeth Boese



Queues



Queues

- FIFO
- Like
 - Lines at the bank/checkout/etc.
- Use Cases
 - batch jobs
 - scheduling
 - next caller

Queues

Operations

- isEmpty
- isFull (*esp for array-based implementations*)
- enqueue
- dequeue
- peek (or top)

Queues

Implementations

- Linked-List
- Array

Queues

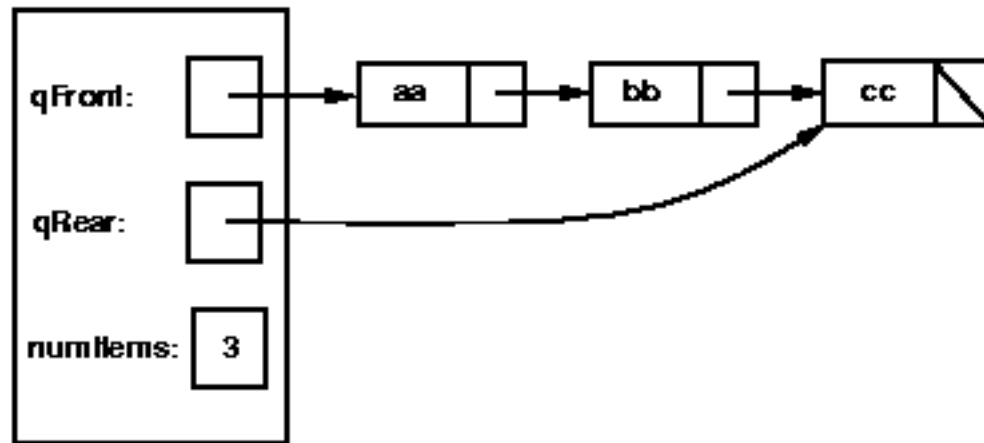
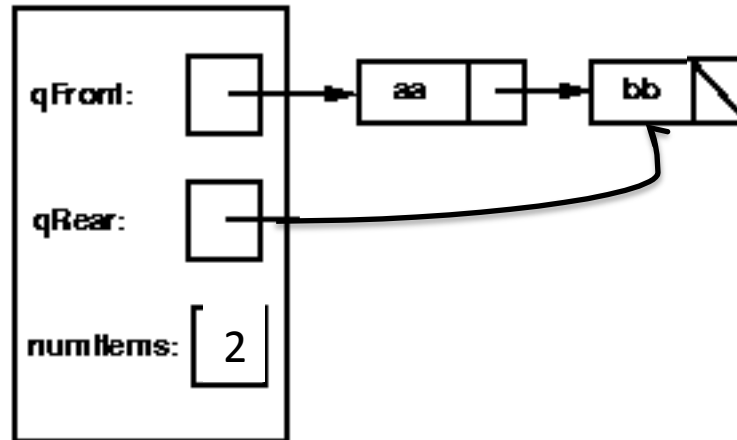
- Linked List

```
private Listnode *ptrFront;  
private Listnode *ptrRear;  
private int numItems;
```

Queues

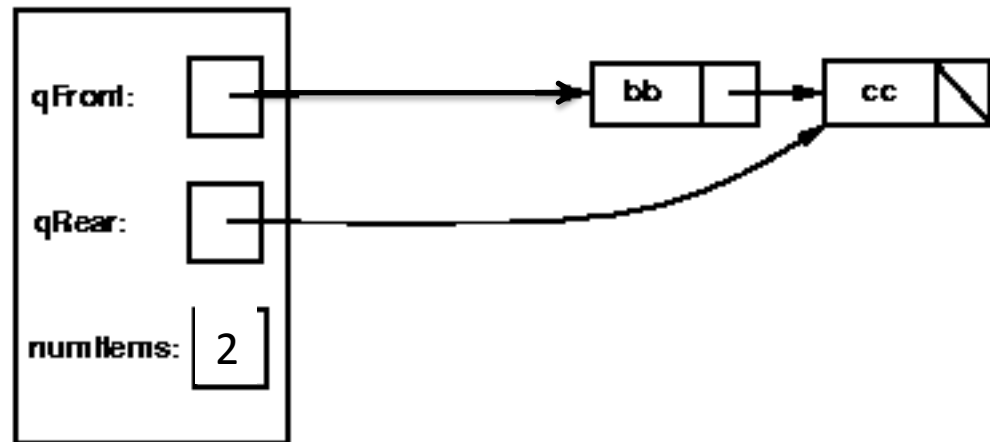
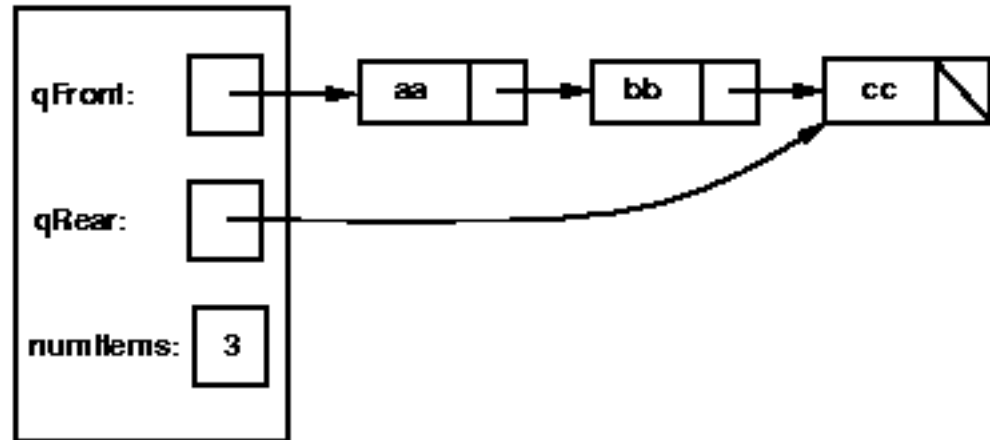
Linked List

– enqueue



Queues

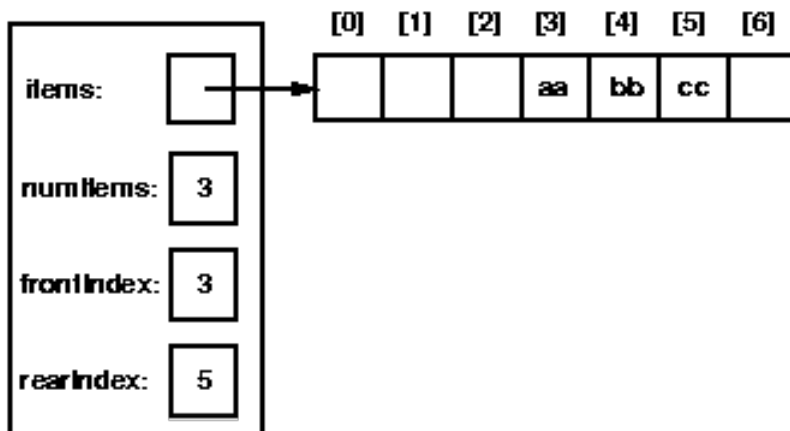
Linked List – dequeue



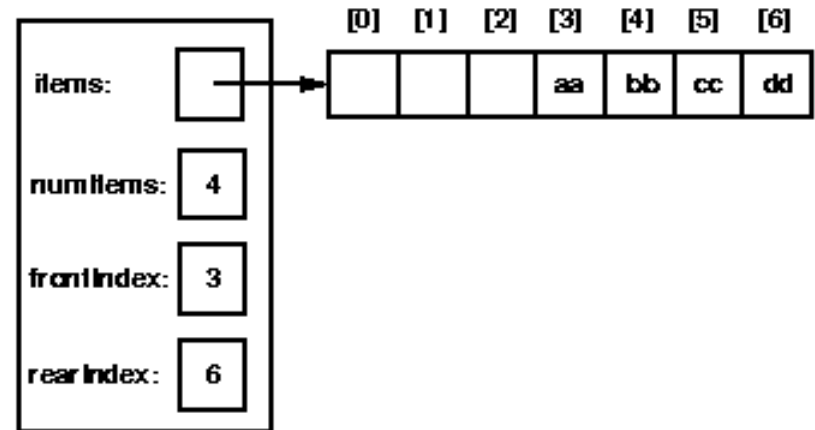
Queues

- Array
 - enqueue

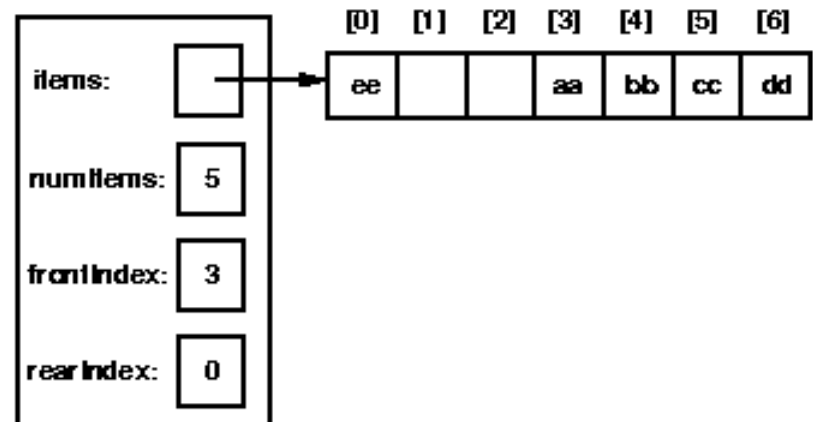
Initial queue:



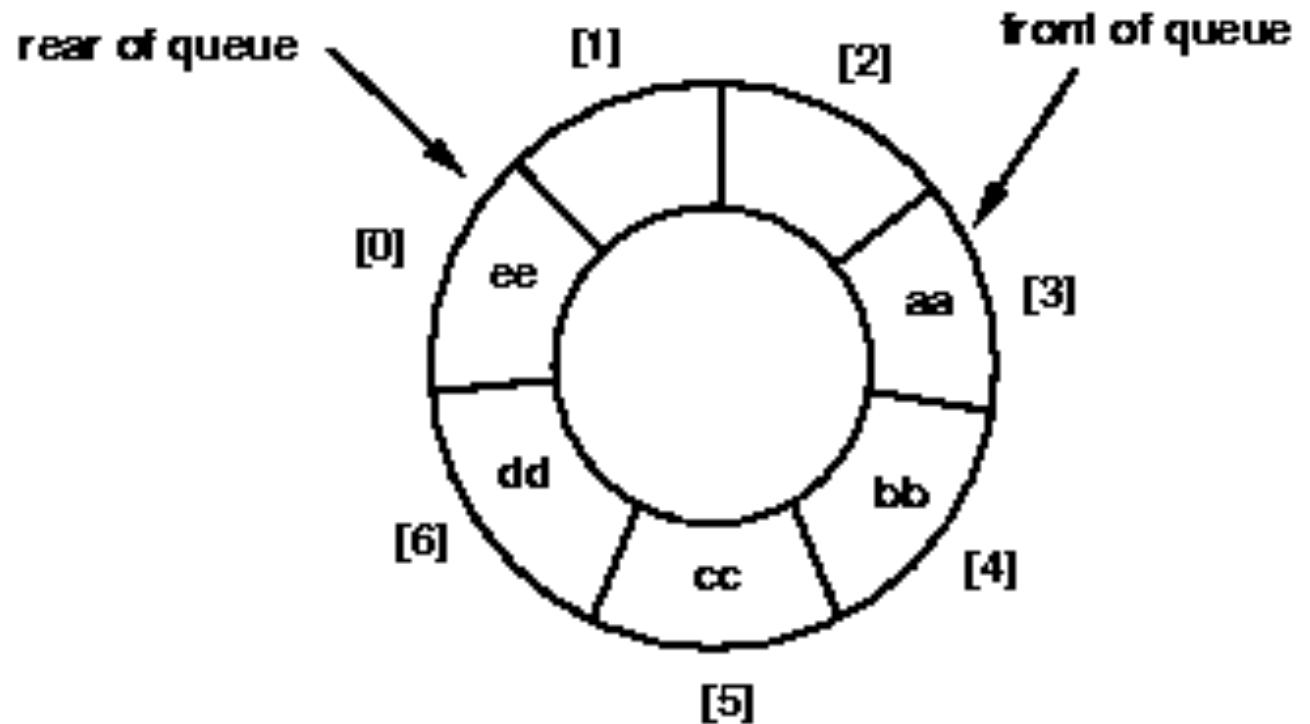
After enqueue(dd):



After enqueue(ee):

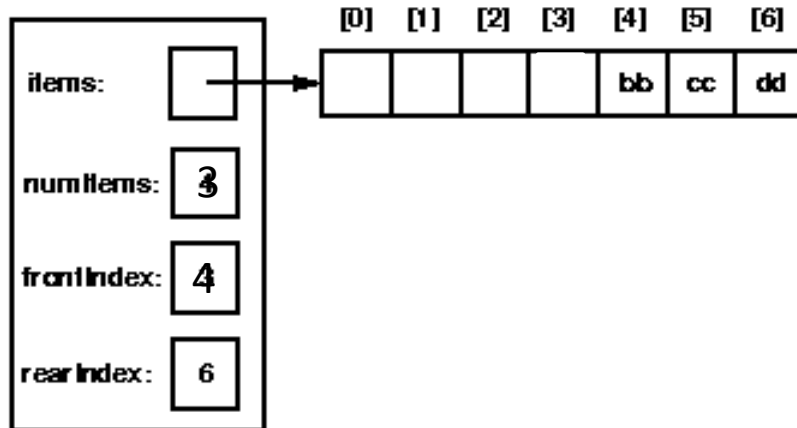
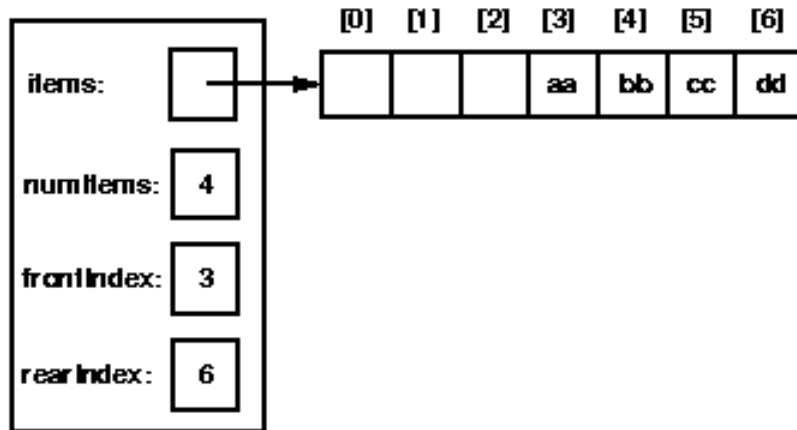


- Array
 - circular



Queues

- Array
 - dequeue



Queues

- Linked-List
 - Fill in the following table, using Big-O notation to give the worst and average-case times for each of the stack methods for a stack of size N.

OPERATION	WORST-CASE TIME	AVERAGE-CASE TIME
isEmpty		
enqueue		
dequeue		
peek		

Queues

- Array
 - Fill in the following table, using Big-O notation to give the worst and average-case times for each of the stack methods for a stack of size N.

OPERATION	WORST-CASE TIME	AVERAGE-CASE TIME
isEmpty		
enqueue		
dequeue		
peek		

Queues



QUESTIONS TO PONDER

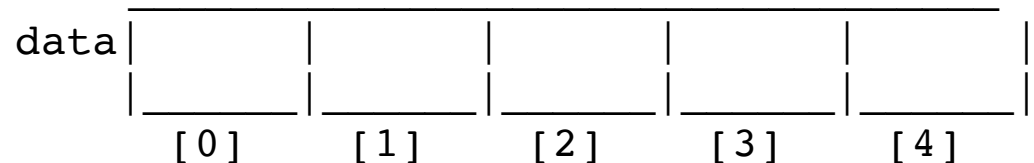
Questions

Execute this code:

```
queue<int> data;  
data.enqueue(1);  
data.enqueue(2);  
data.enqueue(3);  
data.dequeue( );
```

Suppose that data is represented by a circular array.

Draw the state of the private member variables of data after the above code:



Questions

- Execute this code:

```
queue<int> data;  
data.enqueue(1);  
data.enqueue(2);  
data.enqueue(3);  
cout << data.peek( );
```

Suppose that data is represented by a linked list.

Draw the state of the private member variables of data after the above code:

head_ptr |
rear_ptr |

Questions

1. One difference between a queue and a stack is:
 - A. Queues require dynamic memory, but stacks do not.
 - B. Stacks require dynamic memory, but queues do not.
 - C. Queues use two ends of the structure; stacks use only one.
 - D. Stacks use two ends of the structure, queues use only one.
2. If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?
3. Suppose we have a circular array implementation of the queue class, with ten items in the queue stored at data[2] through data[11]. The CAPACITY is 42. Where does the push member function place the new entry in the array?

Questions

1. In the linked list implementation of the queue class, where does the push member function place the new entry on the linked list?
 - A. At the head
 - B. At the tail
 - C. After all other entries that are greater than the new entry.
 - D. After all other entries that are smaller than the new entry.
2. In the circular array version of the queue class (with a fixed-sized array), which operations require linear time for their worst-case behavior?
 - A. front
 - B. push
 - C. empty
 - D. None of these operations require linear time.
3. Consider the implementation of the queue using a circular array. What goes wrong if we try to keep all the items at the front of a partially-filled array (so that `data[0]` is always the front).
 - A. The constructor would require linear time.
 - B. The `get_front` function would require linear time.
 - C. The insert function would require linear time.
 - D. The `is_empty` function would require linear time.