

Lecture Chapter 8 Queues

Queue

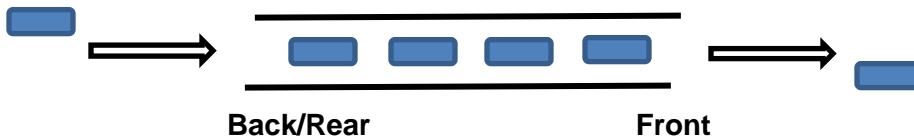
- FIFO
- Front of the Queue
- Back/Rear of the Queue

Stack

- LIFO
- Top of the Stack
- Bottom of the Stack

ADT Queue Operations

- Create Empty Queue
- Determine whether a queue is empty
- Add a new item to a queue
- Remove the item which is at the front of the queue
- Remove all items from the queue
- Retrieve the item which is at the front of the queue
 - retrieves a copy of the item for inspection
 - leaves the item at the front of the queue



Queue
-- front -- back + item
+createQueue () +isEmpty () : Boolean {query} +enqueue(in newItem:QueryItemType) throws QueueException +dequeue() : QueryItemType throws QueueException +dequeueAll () +peek () : QueryItemType {query} throws QueueException

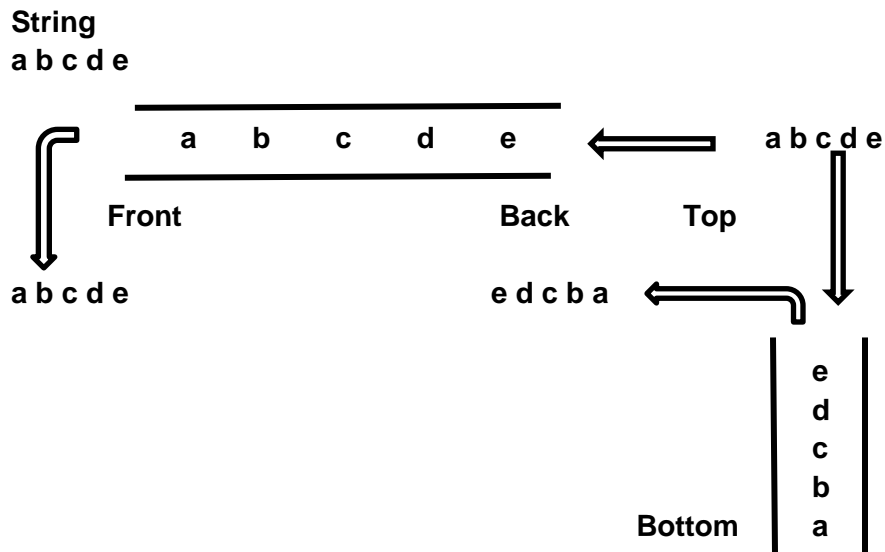
ADT Queue Applications

- Capturing and Holding a Character String (keyboard entry)

```
aQueue.createQueue( )  
while ( ! end of line )  
{  
  // read character ch, e.g., from keyboard  
  aQueue.enqueue (ch)  
}
```

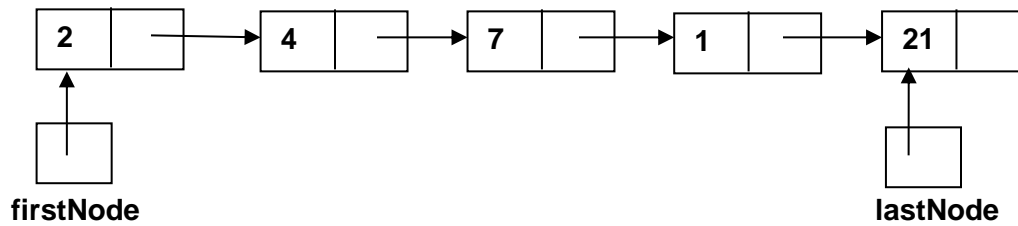
See pages 412-413 for processing arithmetic string captured in the aQueue

- Recognizing Palindromes

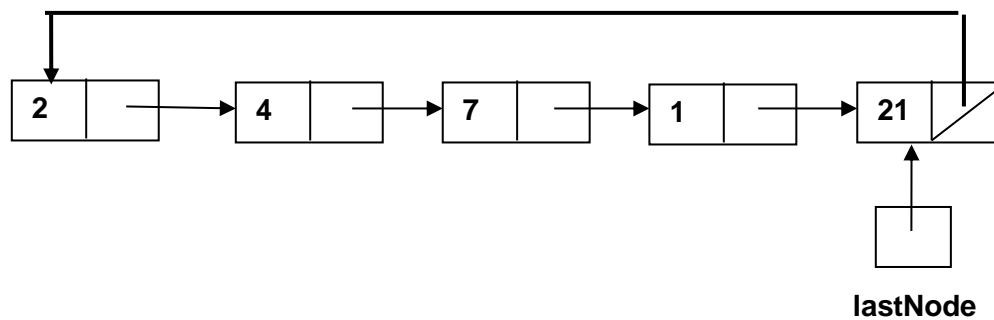


Reference-Based Implementation

linked list with two reference variables

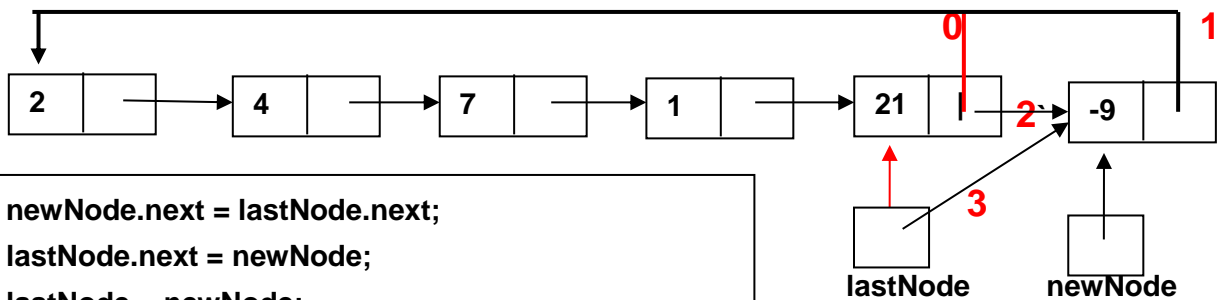


circular linked list



lastNode references the last node in the queue
lastNode.next references the first node in the queue

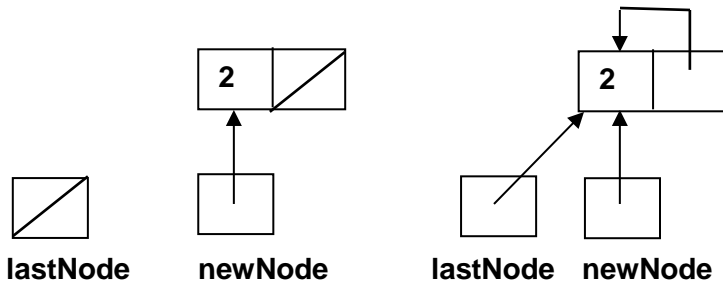
inserting a new node a circular linked list



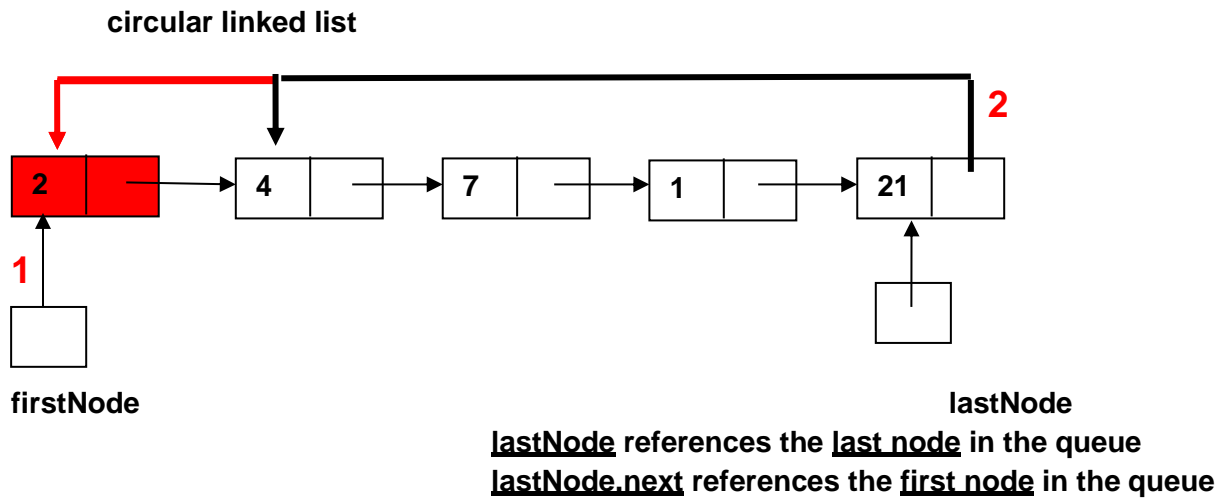
- 1 `newNode.next = lastNode.next;`
- 2 `lastNode.next = newNode;`
- 3 `lastNode = newNode;`

red lines indicate reference values before the changes

inserting the first item into an empty queue



deleting an item from a queue of more than one item



```

1 firstNode = lastNode.next;
2 lastNode.next = firstNode.next;

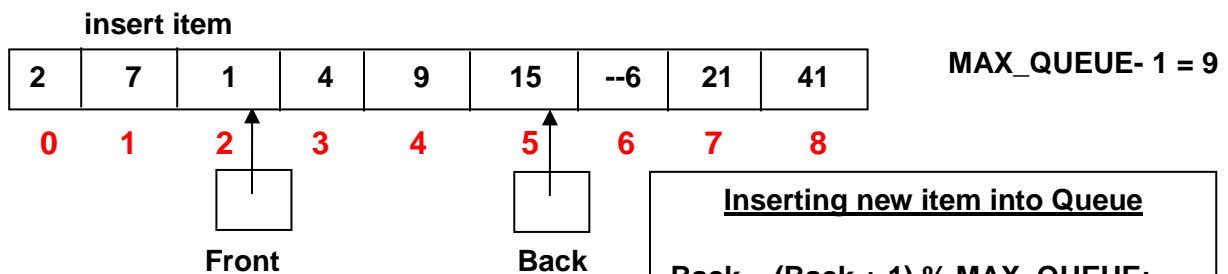
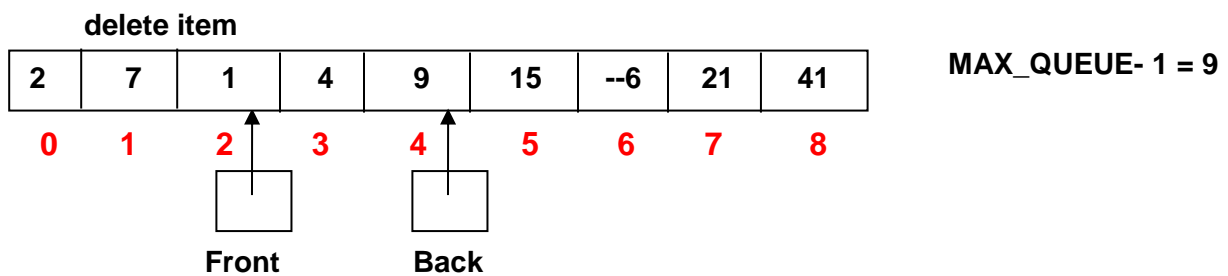
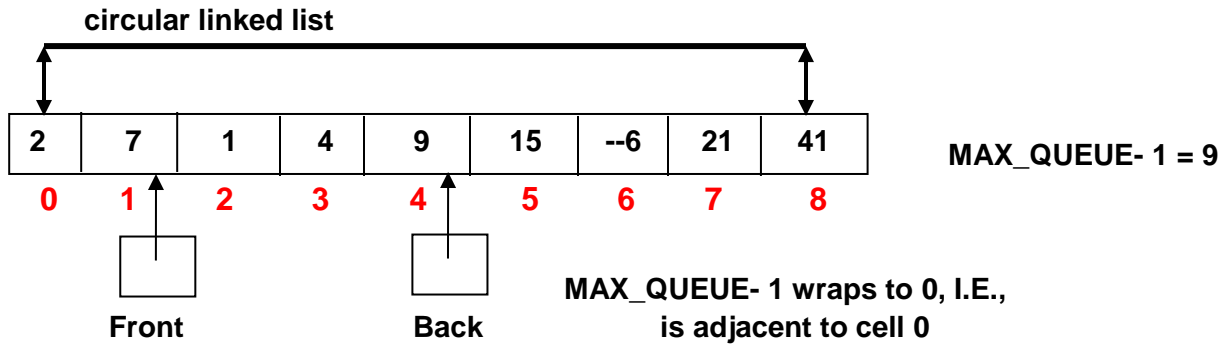
red lines indicate reference values before the changes
    
```

See pages 418-419 for reference-based implementation program

Array-Based Implementation

Circular Array

- Moves clockwise
- Index front to delete an item
- Index back to insert an item



initialize the original queue
 int count = 0;
 Back = QUEUE - 1;
 Front = 0;
 if Front passes Back then
 Queue is either full or empty

Inserting new item into Queue
 Back = (Back + 1) % MAX_QUEUE;
 items [Back] = newItem;
 ++count;

Deleting item from the Front of the Queue
 Front = (Front + 1) % MAX_QUEUE;
 -- count;

See pages 422-424 for array-based implementation program & variations

See pages 425-445 for ADT List, JCF Queue & Deque, comparisons, simulation of bank queues