Queues

- Organizing data
- Queue interface and implementations
- Queue Class
- Examples and applications

Organization of Data

- Organize data by position
  - stacks
  - queues
- Organize data by its value
  - lists (arrays, vectors)
  - trees
  - priority queues

Queue as Data Structures

- To simulate the behavior of phenomena where insertions and deletions occur at opposite ends of the data structure
  - lines of people
  - any sort of line
- FIFO -- first item in is the first out

Queue Operations

- initialize the queue
- is the queue empty?
- is the queue full?
- inserting an item
- deleting an item
- retrieving an item

Queue interface

- boolean isEmpty()
- void enqueue(E obj)
  - <u>inserts</u> object at the back of the queue
- E dequeue()
  - <u>deletes</u> and <u>returns</u> object at the front of the queue
- E peek()
  - <u>retrieves</u> object at the front of the queue

Queue Implementation

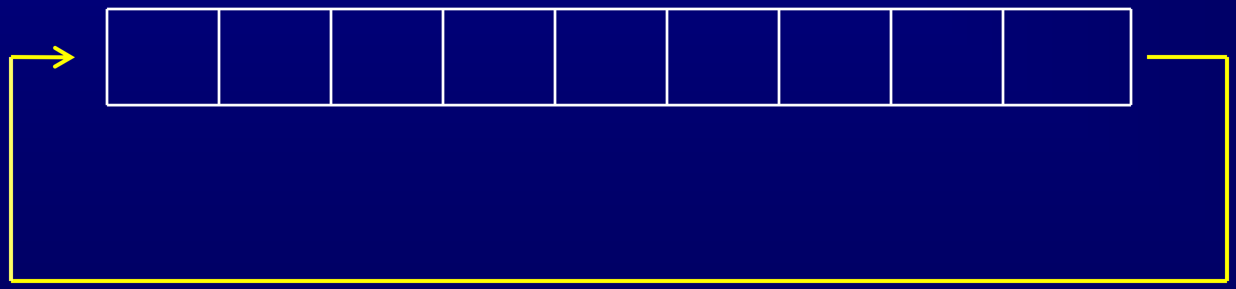
- Linear implementation
  - array-based
  - reference-based
  - pros and cons

- Circular implementation
  - array-based
  - reference-based
  - pros and cons

Queue using arrays

- Store the elements from front to rear
  - Do we store *front* always at index 0?
  - Do we store *rear* always at index length-1?
  - The *rightward drift* problem

- Using a circular array



Queues Applications

- Simulation of lines
  - grocery store
  - banks
  - emergency room
  - printing jobs
  - airports
  - Internet routers

- Priority queues (PQs)
  - Heaps (later on)

Queue Class in Java

- Java provides an *interface* for queues
  - Look up the Queue<E> interface

Homework

- Homework #6 (Stacks & Queues) due on 10/27
- Quiz #7 (Stacks & Queues) on 10/28