Queue ADT

sections
4.6-4.9
4.15-4.16

Queue

- First In First Out (FIFO)
- Must maintain entry order
- Enter one side, exit the other

Queue ADT

```
typedef struct QueueElement* QueueElementPtr;

typedef struct QueueElement {
  int data;
  QueueElementPtr psNext;
} QueueElement;

typedef struct Queue* QueuePtr;

typedef struct Queue {
  QueueElementPtr psEnter;
  QueueElementPtr psExit;
} Queue;
```

Operations

- create(QueuePtr)
- terminate(QueuePtr)
- isFull(QueuePtr)
- isEmpty(QueuePtr)
- size(QueuePtr)

Operations

- enqueue(QueuePtr, Data)
- dequeue(QueuePtr, DataPtr)
- peek(QueuePtr, DataPtr)

Implementation

- Using a singly linked list
  - benefits?
  - disadvantages?
- C struct

- queueCreate, queueEnqueue, queueDequeue

Code, enqueue, dequeue

Code, enqueue, dequeue

Queue ADT

```
typedef struct Queue* QueuePtr;

typedef struct Queue {
	ListPtr psList;
} Queue;
```

// We say the Queue is ***backed*** by a List.

Code, enqueue, dequeue

Code, enqueue, dequeue

Implementation

- An array
  - circular vs not
- What does struct Queue look like?
- Advantages/disadvantages?
  - challenges?
  - queueCreate, queueIsFull, queueEnqueue, queueDequeue

Code

Implementation

- Using a doubly linked circular list
  - benefits?
  - disadvantages?
- C struct

- queueCreate, queueEnqueue, queueDequeue

Code