CSE 3010 – Data Structures & Algorithms

Lecture #21

What will be covered today

- Understanding first-in-first-out structure
- Algebraic specification for Queue ADT
- Applications using Queue
- Implementing Queue using arrays

Algebraic specification of Queue ADT – Types and Operations

Types:

`QUEUE, ITEM, BOOLEAN`

where `ITEM` is of some type the `QUEUE` will contain

`BOOLEAN` is either `True` or `False`

Operations:

```
create:                 -> QUEUE
add: QUEUE x ITEM       -> QUEUE
remove: QUEUE           -> QUEUE
front: QUEUE            -> ITEM
isEmpty: QUEUE          -> BOOLEAN
```

Algebraic specification of Queue ADT - Axioms

For all Q ∈ QUEUE, **and** k ∈ ITEM

```
remove(create()) = error
remove(add(Q,k)) = add(remove(Q),k)

front(create())  = error
front(add(Q,k)) = front(Q)

isEmpty(create()) = true
isEmpty(add(Q,k)) = false
```

QUEUE is a container of elements with first-in first-out (FIFO) data structure

Applications using Queue data structure

- Scheduling jobs
- Breadth-first search
- Simulation of applications requiring FIFO
- Page replacement algorithm in operating systems

Implementing Queue using arrays

front = -1
rear = -1

Empty queue

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Points to note about static implementation (using array) of a queue:

- Size of queue (array) is fixed
- Queue is empty when created
- front and rear of the queue set to -1 when queue is empty
- Queue is always empty when front = -1