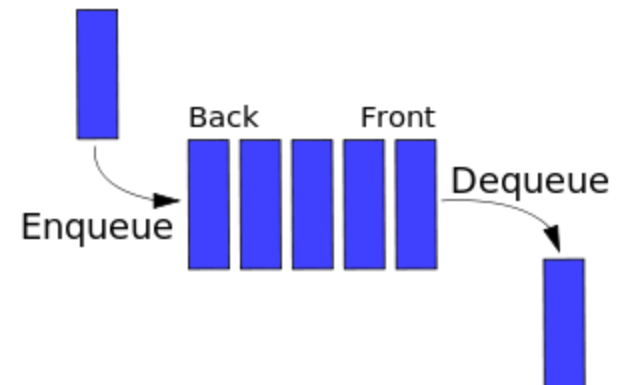


QUEUES

Queue

- The queue is a FIFO (First-in First-out) data structure
- Elements are added at the rear of the queue and removed from the front
- The only data element that can be removed is the least recently added element



Queue ADT: Specification

- Elements:
 - Queue elements can be of any type, but we will assume QueueElement
- Structure:
 - Any mechanism for determining the elements order of arrival into the queue

Queue ADT Continued

- **Domain:**
 - The number of queue elements is bounded. A queue is considered full if the upper-bound is reached. A queue with no elements is considered empty.

Queue ADT Continued

Operations: There are six operations as follows:

- function create (q: Queue, isCreated: boolean)
 - **results:** if q cannot be created, isCreated is false; otherwise, isCreated is true, the queue is created and is empty
- function terminate (q: Queue)
 - **results:** queue q no longer exists

Queue ADT Continued

- function `isFull (q: Queue)`
 - **results:** returns true if the queue is full; otherwise false is returned
- function `isEmpty (q: Queue)`
 - **results:** returns true if the queue is empty; otherwise, false is returned
- function `enqueue (q: Queue, e: QueueElement)`
 - **requires:** `isFull (q)` is false
 - **results:** element `e` is added to the rear of the queue as the most recently added element

Queue ADT Continued

- function dequeue (q: Queue, e: QueueElement)
 - **requires:** isEmpty(q) is false
 - **results:** The least recently added element is removed from the front of the queue and assigned to e

Queue Implementation

- How would you implement a queue?

Queue Implementation

- Using an array
 - How would this work?
 - What are the drawbacks?
- Define the data structure in C
- Write `qCreate`, `qIsFull`, `qEnqueue`

Queue Implementation

- Using an array (circular queue)
 - How would this work?
 - What are the drawbacks?
- Define the data structure in C
- Write `qCreate`, `qIsFull`, `qEnqueue`

Queue Implementation

- Using a singly linked list
 - How would this work?
 - What are the drawbacks?
- Define the data structure in C
- Write `qCreate`, `qIsFull`, `qEnqueue`

Queue Implementation

- Using a singly linked circular list
 - How would this work?
 - What are the drawbacks?
- Define the data structure in C
- Write `qCreate`, `qIsFull`, `qEnqueue`