

## Implementation of a stack with an array

### Variables

*stack*: array of elements

*top*: integer

invariant:  $stack[0], \dots, stack[top]$  are the elements of the stack listed from bottom to top

### Initialization

$top \leftarrow -1$

### Algorithms

size():

*output*: size of stack

**return** ( $top + 1$ )

isEmpty():

*output*: stack is empty?

**return** ( $top < 0$ )

top():

*precondition*: stack is nonempty

*output*: top element of stack

**return**  $stack[top]$

push(*element*):

*precondition*: stack is not full

*postcondition*: *element* has been added onto top of stack

*input*: *element* to be added to stack

$top \leftarrow top + 1$

$stack[top] \leftarrow element$

pop():

*precondition*: stack is nonempty

*postcondition*: top element has been removed from stack

*output*: top element of stack

$temp \leftarrow stack[top]$

$top \leftarrow top - 1$

**return** *temp*

## Implementation of a queue with a circular array

### Variables

*queue*: array of elements

*front*: integer

*rear*: integer

*capacity*: integer

invariant: if  $front \leq rear$ , then  $queue[front], \dots, queue[rear - 1]$  are the elements of the queue from front to rear; otherwise,  $queue[front], \dots, queue[capacity - 1], queue[0], \dots, queue[rear - 1]$  are the elements of the queue from front to rear

## Initialization

$front \leftarrow 0$   
 $rear \leftarrow 0$   
 $capacity \leftarrow$  capacity of the array

## Algorithms

size():

*output:* size of queue  
**return**  $(capacity - front + rear) \bmod capacity$

isEmpty():

*output:* queue is empty?  
**return**  $(front = rear)$

front():

*precondition:* queue is nonempty  
*output:* front element of queue  
**return**  $queue[front]$

enqueue(*element*):

*precondition:* array *queue* holds less than  $capacity - 1$  elements  
*postcondition:* *element* has been added at the rear of queue  
*input:* *element* to be added to queue  
 $queue[rear] \leftarrow element$   
 $rear \leftarrow (rear + 1) \bmod capacity$

dequeue():

*precondition:* queue is nonempty  
*postcondition:* front element has been removed from queue  
*output:* front element of queue  
 $temp \leftarrow queue[front]$   
 $front \leftarrow (front + 1) \bmod capacity$   
**return** *temp*