

The background is a light blue gradient. It is decorated with various abstract geometric shapes in two shades of blue: a medium blue and a darker navy blue. These shapes include circles of different sizes, semi-circles, and quarter-circles, scattered across the corners and edges of the frame. The word "STACK" is centered in a bold, dark blue, sans-serif font.

STACK

STACK

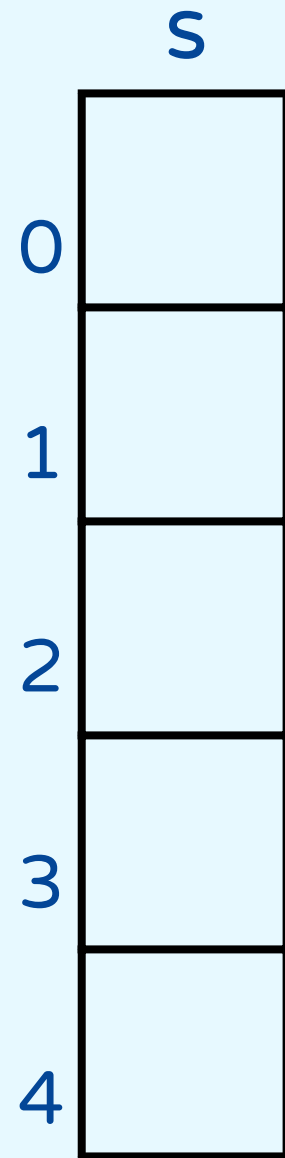
- special kind of list
- insertion and deletion takes place at one end (TOP)
- also known as pushdown

LAST IN, FIRST OUT

FIRST IN, LAST OUT



STACK OPERATIONS



`push(x,s[])` insert First

`pop(s[])` deleteFirst

`top(s[])` returns 1st element

`peek(s[])` looks at top value

`isEmpty(s[])` checks if stack is empty

`isFull(s[])` checks if stack is full

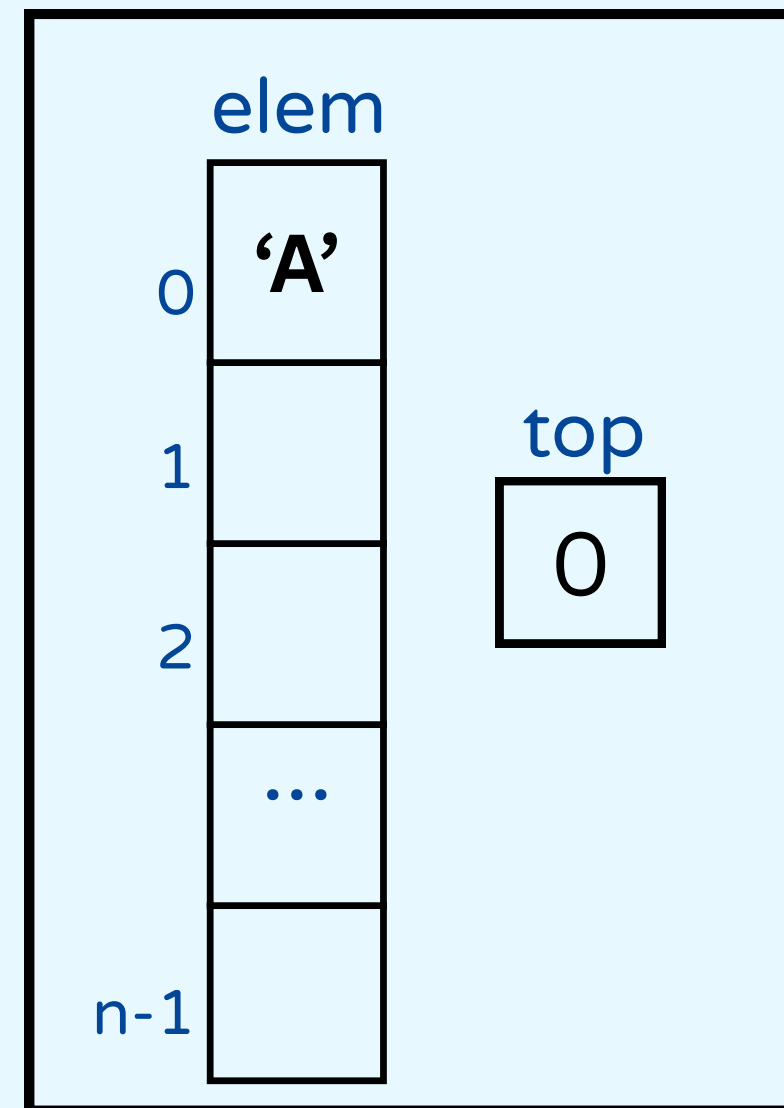
`initStack(s[])` initialize garbage to empty

`makeNULL(s[])` full to NULL

`makeEmpty(s[])` full to empty

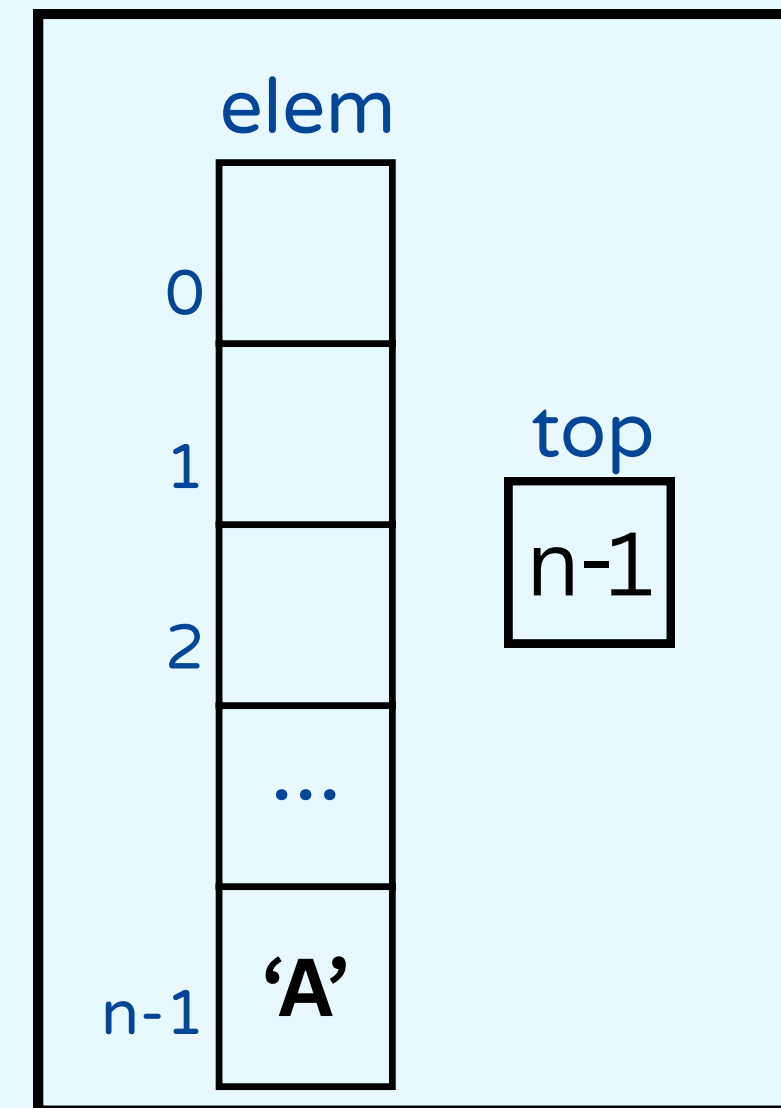
ARRAY IMPLEMENTATION

STACK S



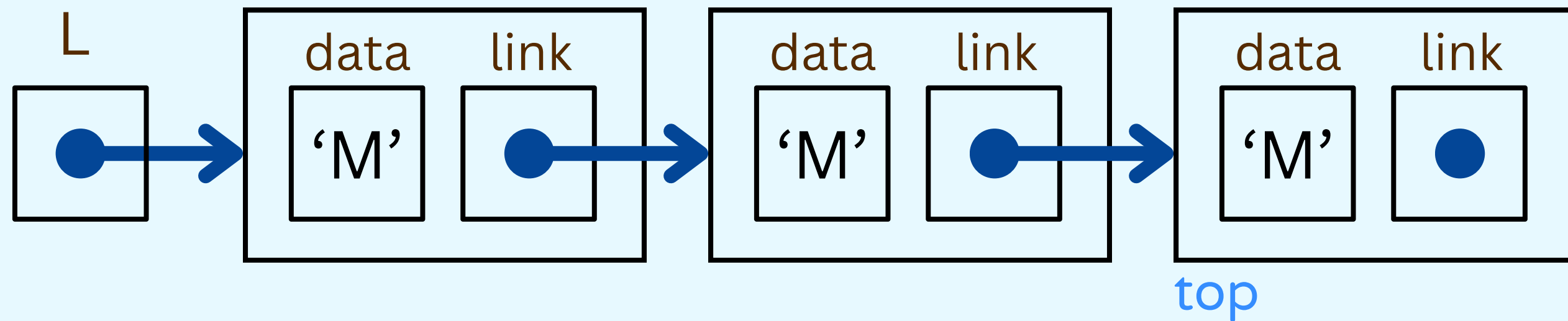
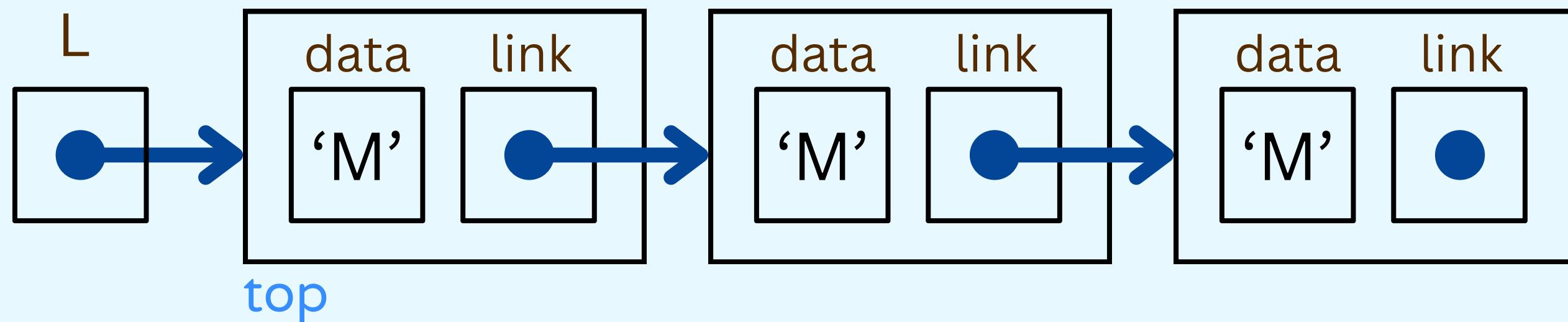
0 to n-1

STACK S



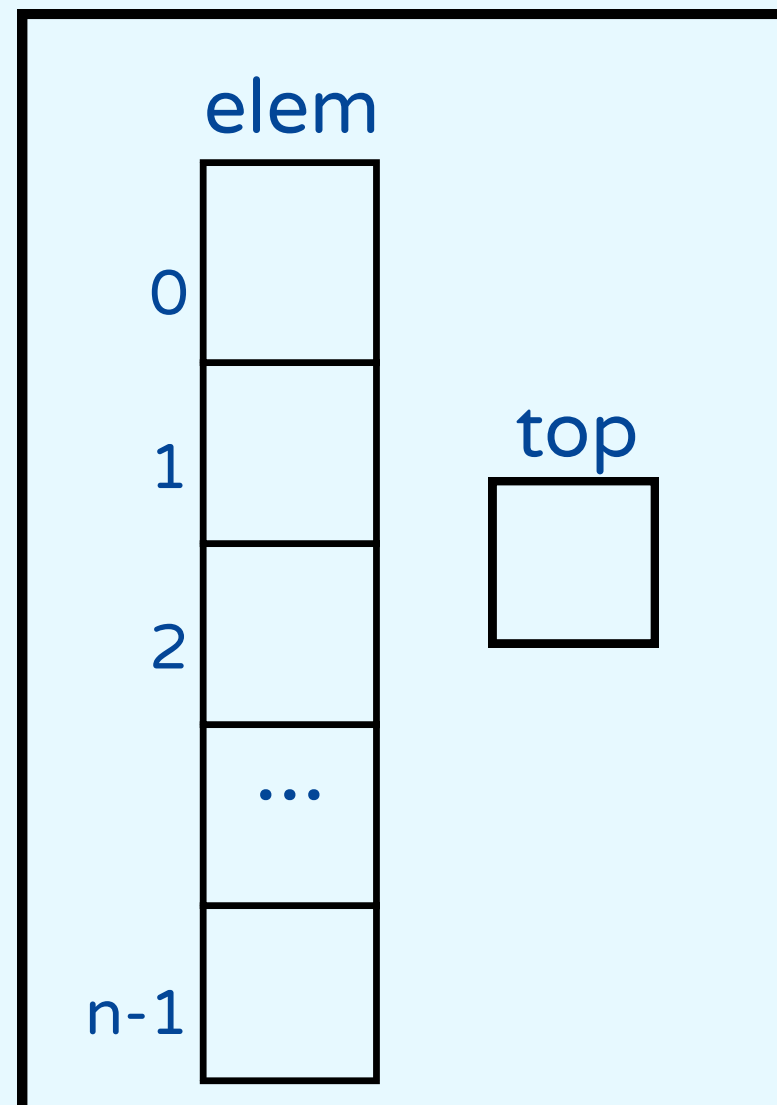
n-1 to 0

LINKED LIST IMPLEMENTATION



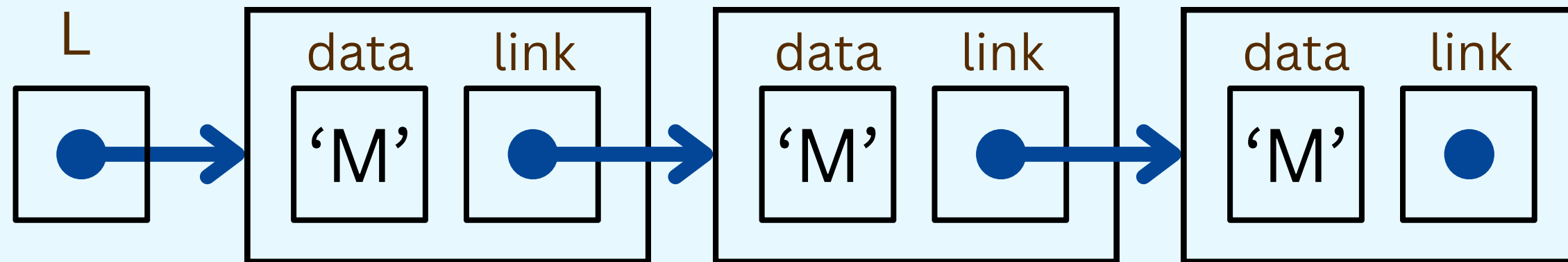
STACK O NOTATION

STACK S



ADT LIST OPERATIONS	STACK OPERATIONS
insertFirst	push $O(n)$
deleteFirst	pop $O(n)$
insertLast	push $O(1)$
deleteLast	pop $O(1)$

STACK O NOTATION



ADT LIST OPERATIONS	STACK OPERATIONS
insertFirst	push $O(1)$
deleteFirst	pop $O(1)$
insertLast	push $O(n)$
deleteLast	pop $O(n)$

The background is a light blue gradient. It is decorated with various geometric shapes in two shades of blue: a medium blue and a darker navy blue. These shapes include circles of different sizes, semi-circles, and quarter-circles, scattered primarily along the top and bottom edges of the frame.

QUEUE

QUEUE

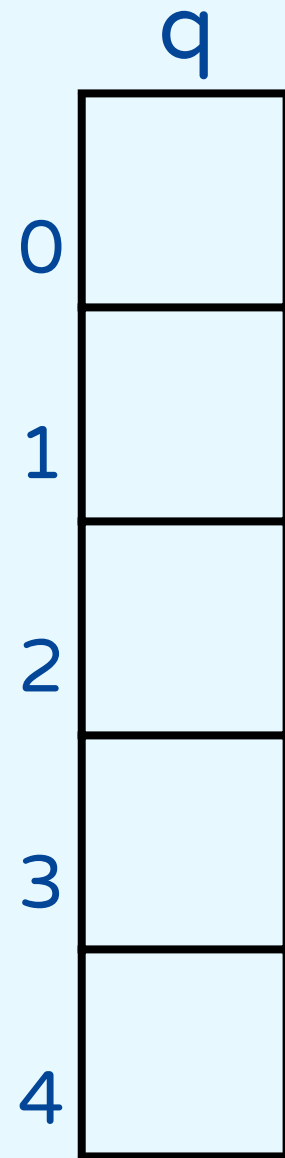
- special kind of list
- does operations on both ends
 - 2 endpoints
- insertion is done on one side, deletion is done on the other (REAR/FRONT)

FIRST IN, FIRST OUT

LAST IN, LAST OUT



QUEUE OPERATIONS



`enqueue(x,q[])` insert rear

`dequeue(q[])` delete front

`front(q[])` returns 1st element

`isEmpty(q[])` checks if queue is empty

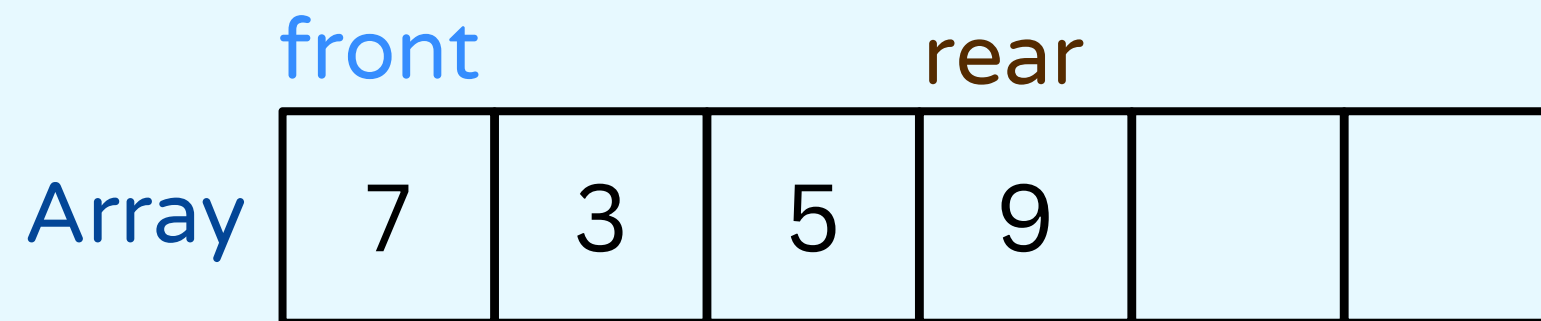
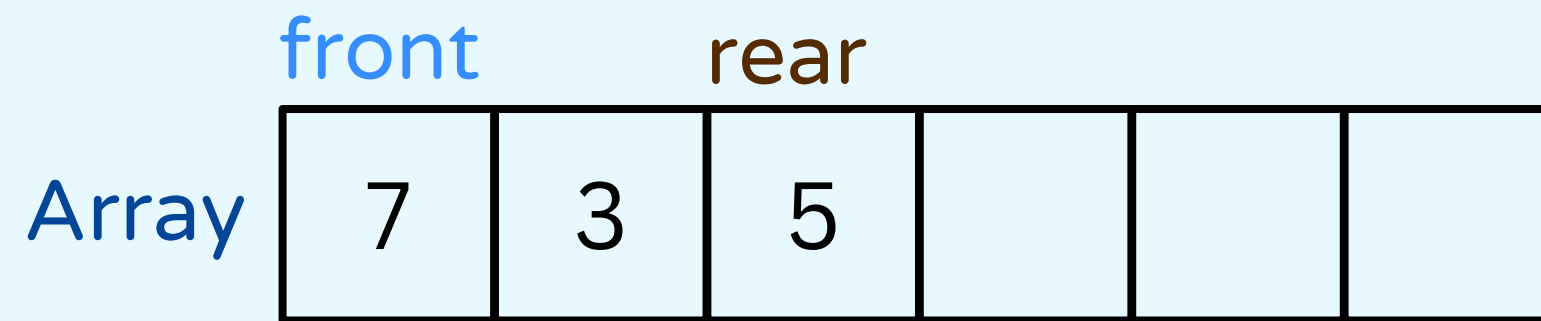
`isFull(q[])` checks if queue is full

`initQueue(q[])` initialize garbage to empty

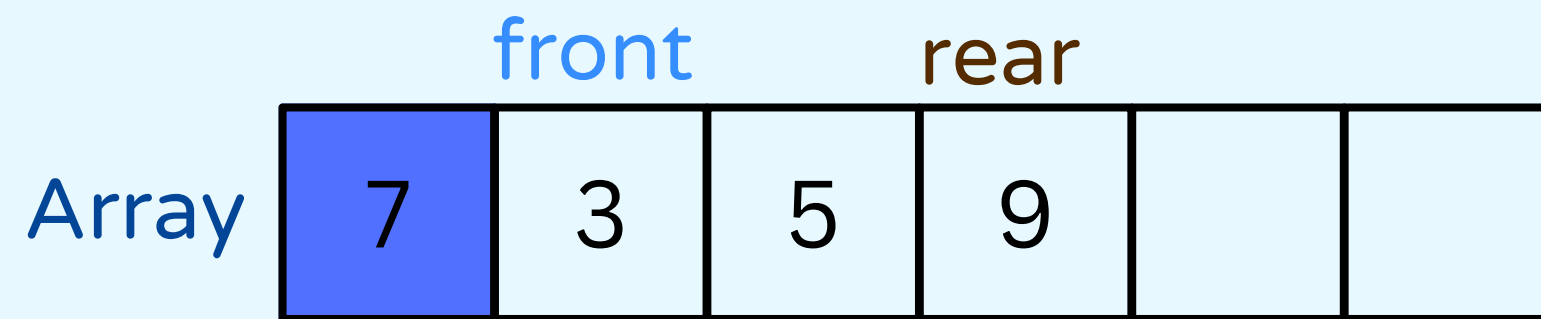
`makeNULL(q[])` full to NULL

`makeEmpty(q[])` full to empty

ARRAY IMPLEMENTATION



enqueue (9)



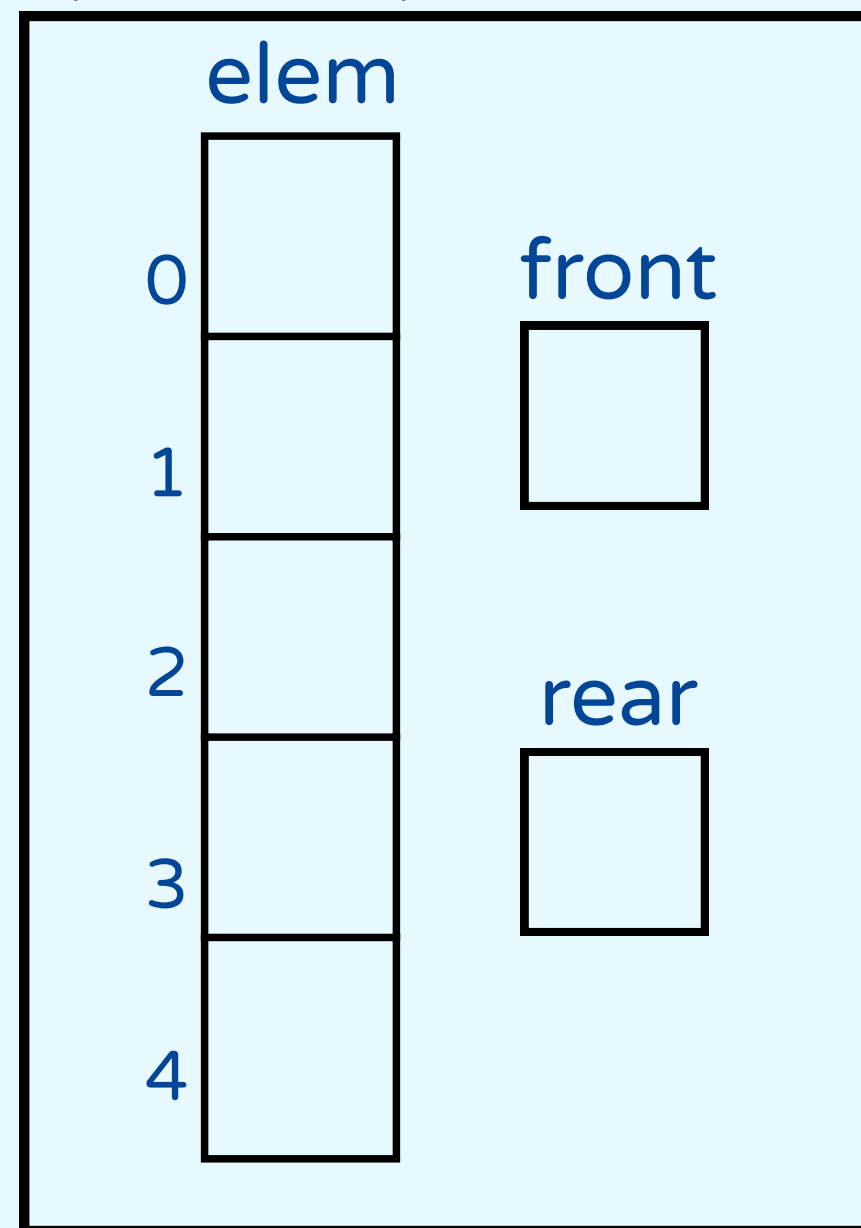
dequeue ()

Problem:

- can't access the first index
- array space for values become limited

ARRAY IMPLEMENTATION

QUEUE Q



Empty

$\text{front} = -1$

$\text{rear} = -1$

Full

$\text{front} = 0$

$\text{rear} = \text{MAX}-1$

front and rear

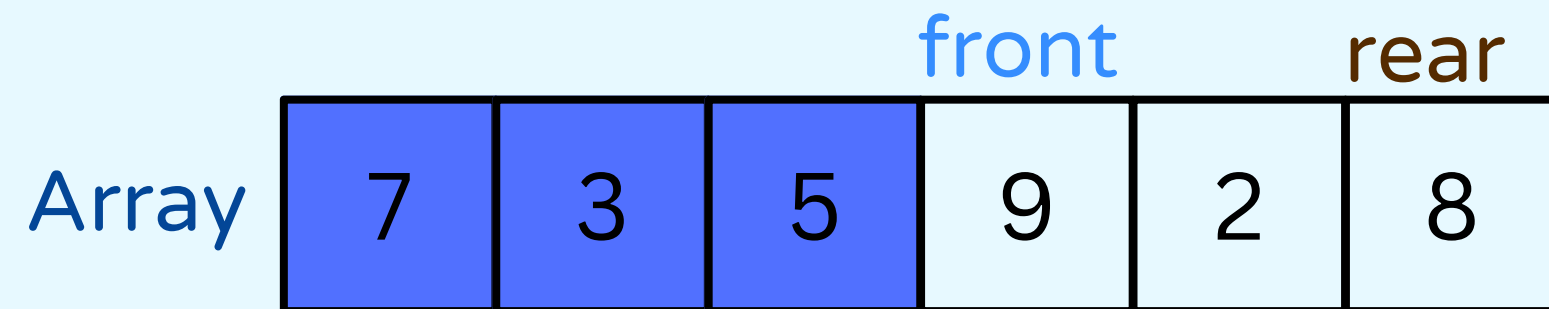
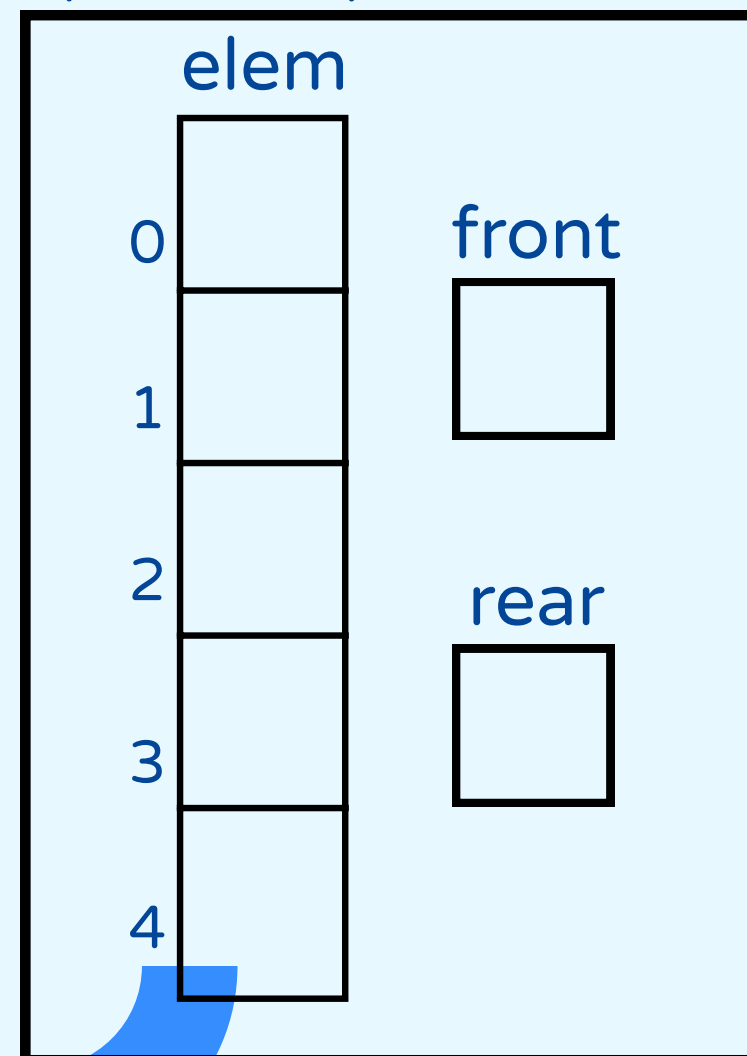
index of array

ARRAY IMPLEMENTATION

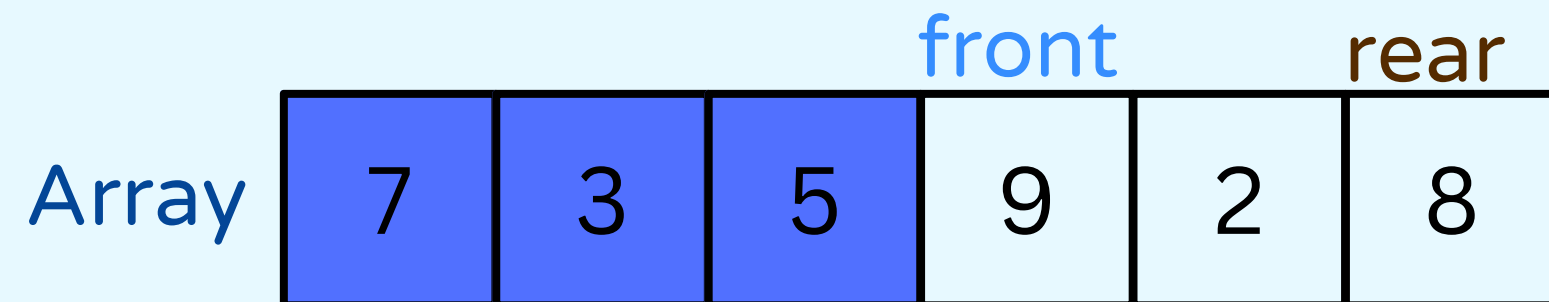
Problem:

- run out of space

QUEUE Q

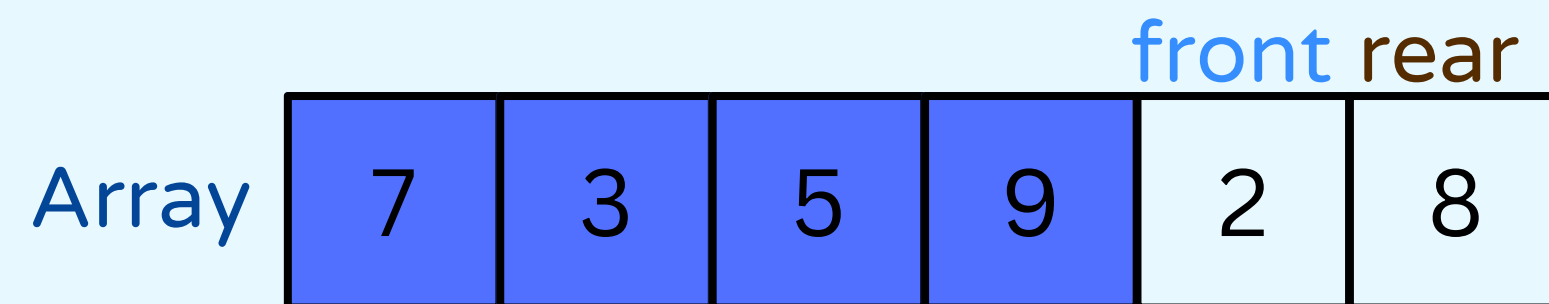


enqueue (8)

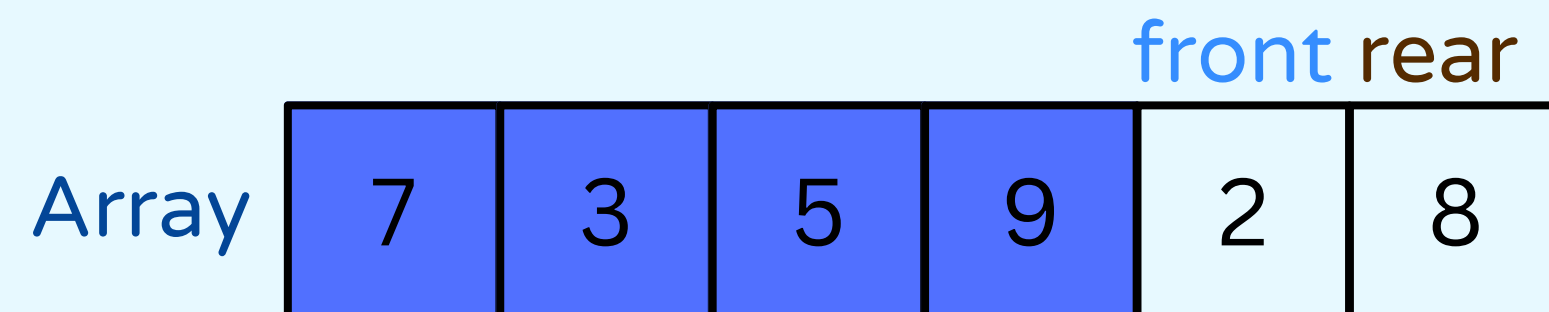


enqueue (1)

error! full!



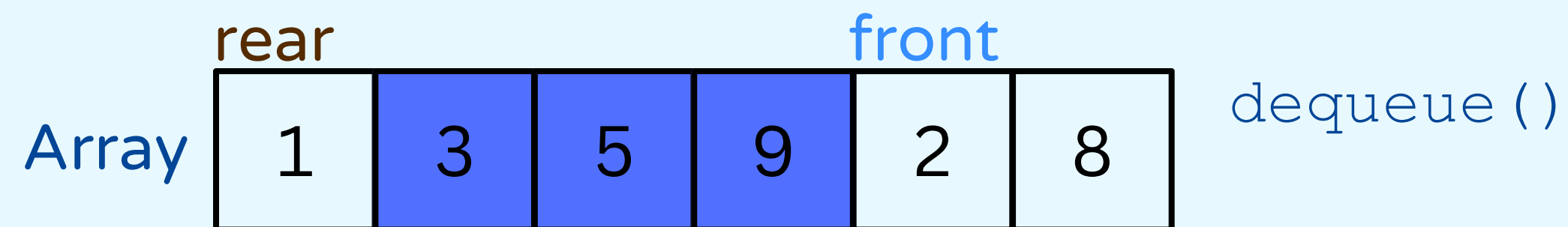
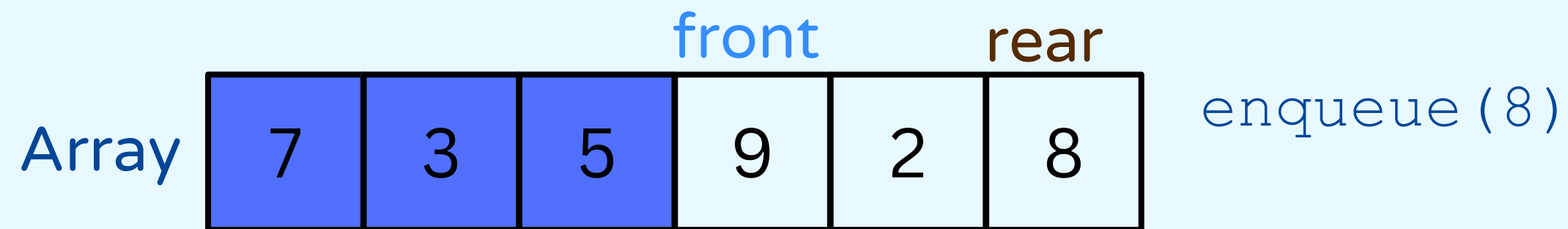
dequeue ()



enqueue (4)

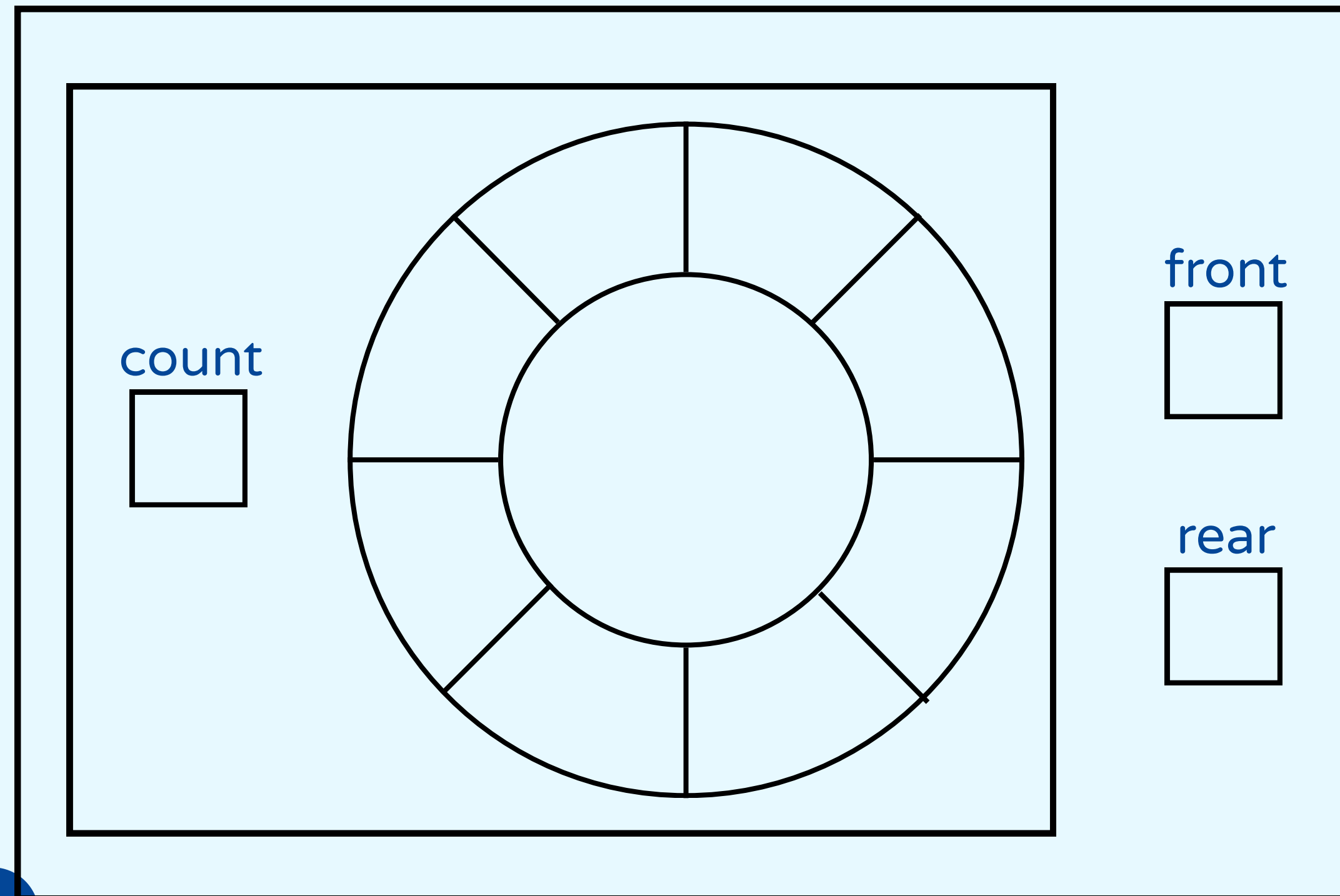
error! full!

SOLUTION: Circular Array



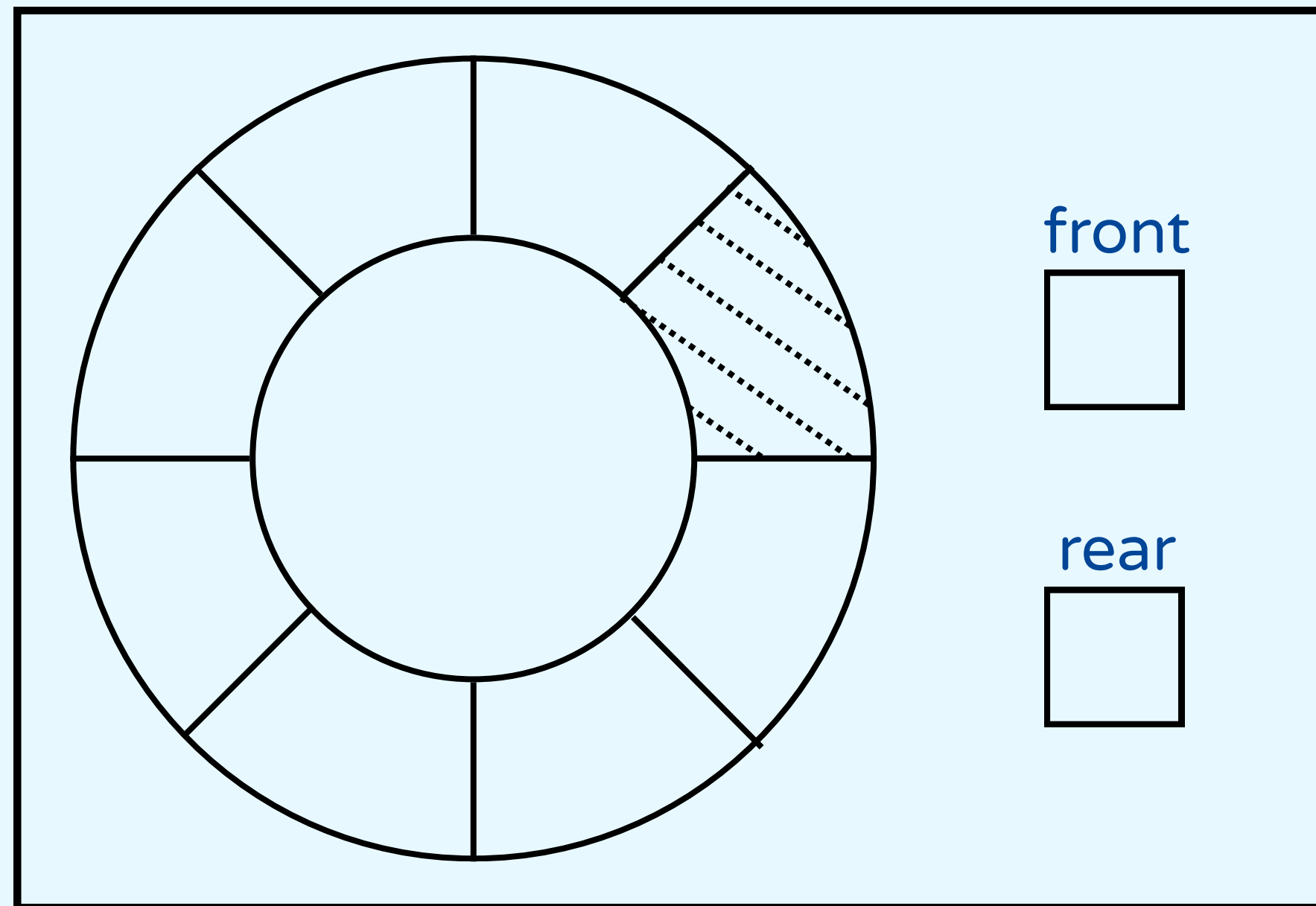
SOLUTION: Circular Array (V1)

QUEUE Q



SOLUTION: Circular Array (V2)

QUEUE Q



OPERATIONS

ADT List	Stack	Queue
<ul style="list-style-type: none">• insertFirst/insertLast• deleteFirst• getValue	<ul style="list-style-type: none">• push• pop• peek/top	<ul style="list-style-type: none">• enqueue• dequeue• front

The background features several abstract geometric shapes in two shades of blue. In the top right, there is a small dark blue circle and a larger light blue semi-circle. On the right side, there is a light blue semi-circle and a dark blue semi-circle. In the bottom right, there is a dark blue circle. On the left side, there is a dark blue circle, a light blue semi-circle, a dark blue semi-circle, and a light blue semi-circle. At the bottom left, there is a dark blue semi-circle and a light blue semi-circle.

**THANK
YOU**