QUEUES

OBJECTIVES

- Define what a queue is
- Understand use cases for a queue
- Implement operations on a queue data structure

WHAT IS A QUEUE?

A **FIFO** data structure!

First **I**n **F**irst **O**ut

WE'VE SEEN THIS BEFORE

Queues exist everywhere! Think about the last time you waited in line....

How do we use them in programming?

- Background tasks
- Uploading resources
- Printing / Task processing

BUILDING A QUEUE WITH AN ARRAY

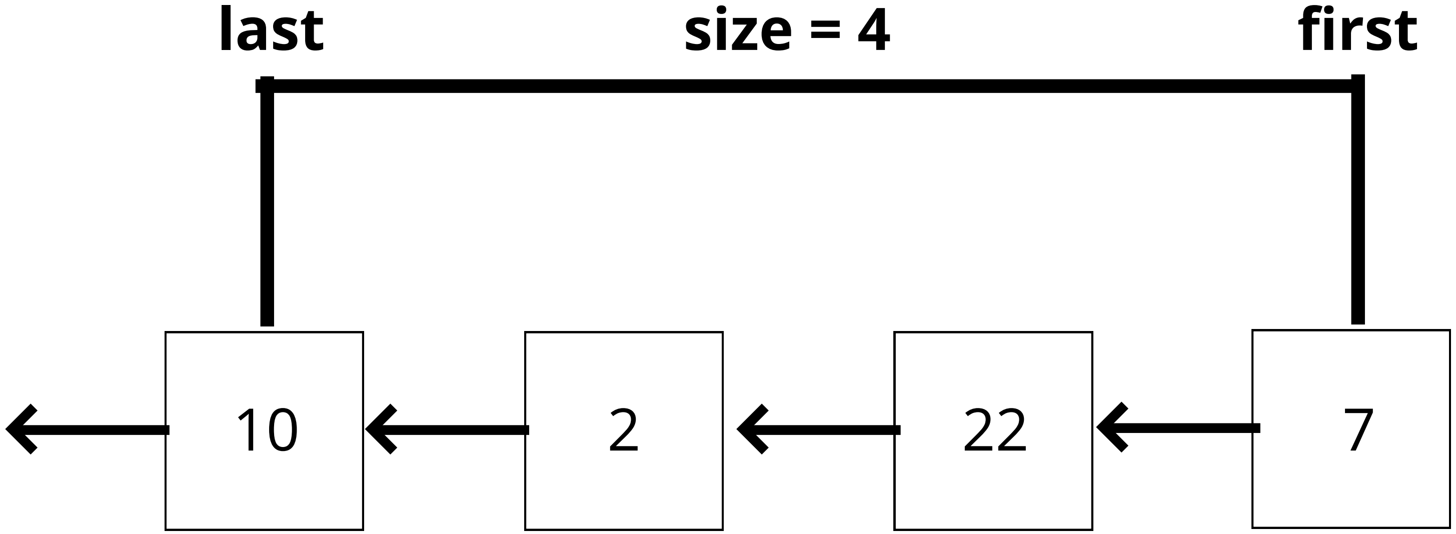
A Queue Class

```
class Queue {
    constructor(){
        this.first = null;
        this.last = null;
        this.size = 0;
    }
}
```

```
class Node {
    constructor(value){
        this.value = value;
        this.next = null;
    }
}
```

How we'll visualize a queue

A series of nodes!



Let's see this in action!

Enqueue

Adding to the **beginning** of the Queue!

Remember, queues are a **FIFO** data structure

Enqueue Pseudocode

- This function accepts some value
- Create a new node using that value passed to the function
- If there are no nodes in the queue, set this node to be the first and last property of the queue
- Otherwise, set the next property on the current last to be that node, and then set the last property of the queue to be that node
- Increment the size of the queue by 1

YOUR TURN

Dequeue

Removing from the **beginning** of the Queue!

Remember, queues are a **FIFO** data structure

Dequeue pseudocode

- If there is no first property, just return null
- Store the first property in a variable
- See if the first is the same as the last (check if there is only 1 node). If so, set the first and last to be null
- If there is more than 1 node, set the first property to be the next property of first
- Decrement the size by 1
- Return the value of the node dequeued

YOUR

TURN

BIG O of QUEUES

Insertion - **O(1)**

Removal - **O(1)**

Searching - **O(N)**

Access - **O(N)**

RECAP

- Queues are a **FIFO** data structure, all elements are first in first out.
- Queues are useful for processing tasks and are foundational for more complex data structures
- Insertion and Removal can be done in **O(1)**