

[Back to Resources](#)

Queues and Stacks

by Jenny Chen, Eddie Huang

Try Me

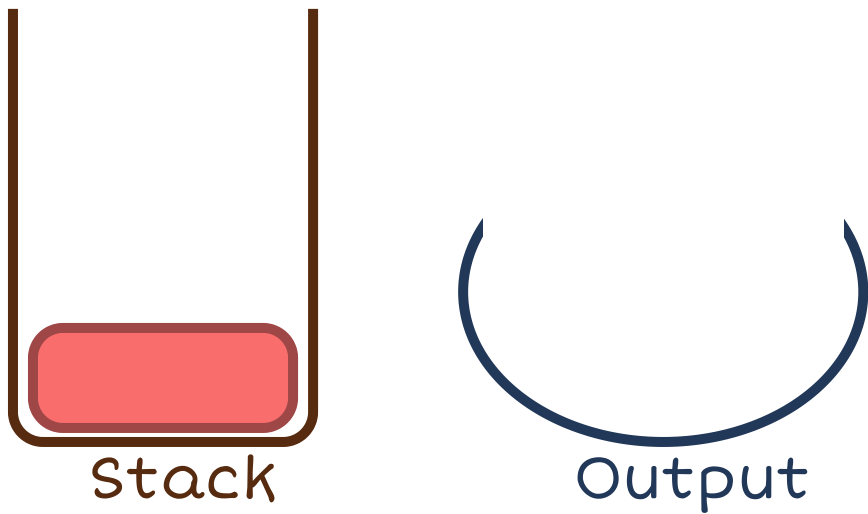
Stack

Queue

Data

Push

Pop



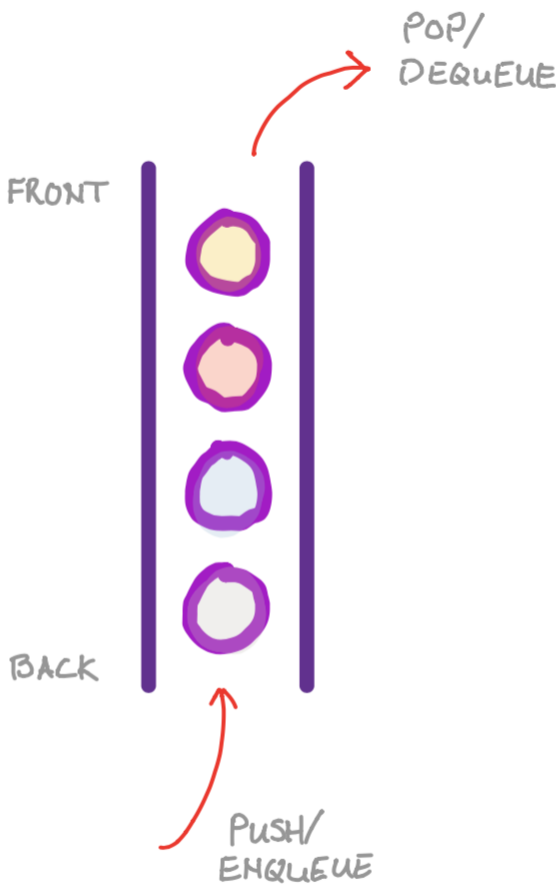
This animation is provided by Jenny Chen and Yanchen Lu

Introduction

Queues and stacks (quacks) are data structures for storing and querying elements in particular linear orders. Queues and stacks share a common interface:

- **push** - adds an element. (For queues, this is also known as **enqueue**)
- **pop** - queries/removes an element. (For queues, this is also known as **dequeue**)

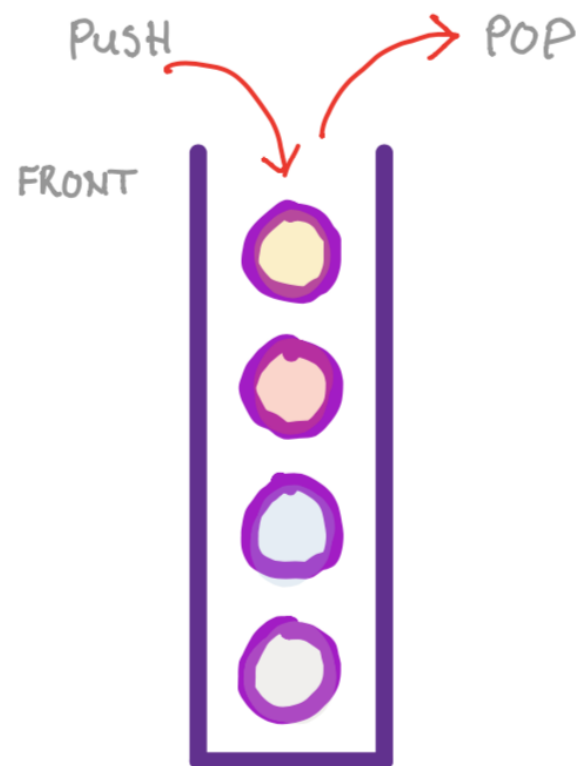
Queues



A queue returns elements in the order in which they were stored.

Queues return elements in the order in which they were stored. We call this kind of data structure *first-in-first-out (FIFO)*.

Stack



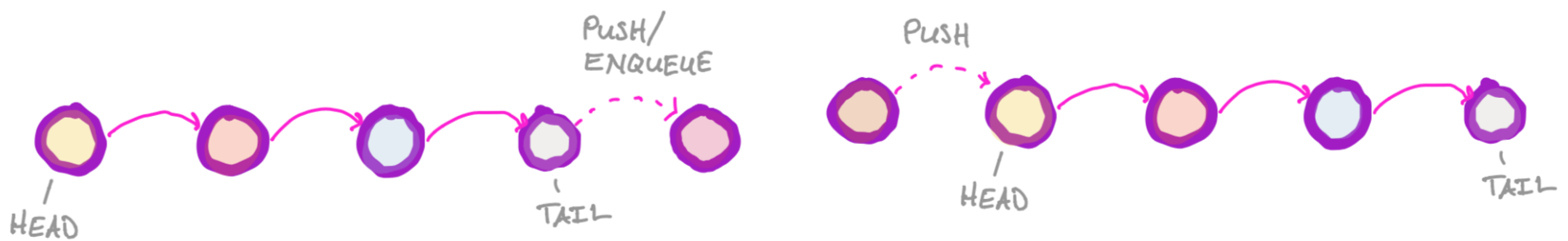
A stack outputs elements in the *reverse* order in which they were stored

Stacks return elements in the reverse order in which they are stored; that is, the most recent element to be added is returned. We call this kind of data structure *last-in-first-out (LIFO)*.

Implementations

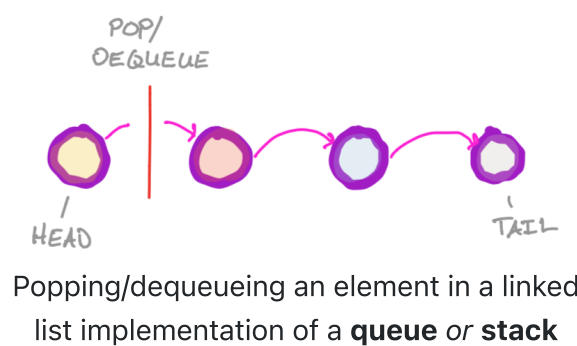
Queues and stacks are often implemented with either linked lists or arrays.

Linked Lists



Pushing an element in a linked list implementation of a **stack**

Enqueueing an element in a linked list implementation of a **queue**



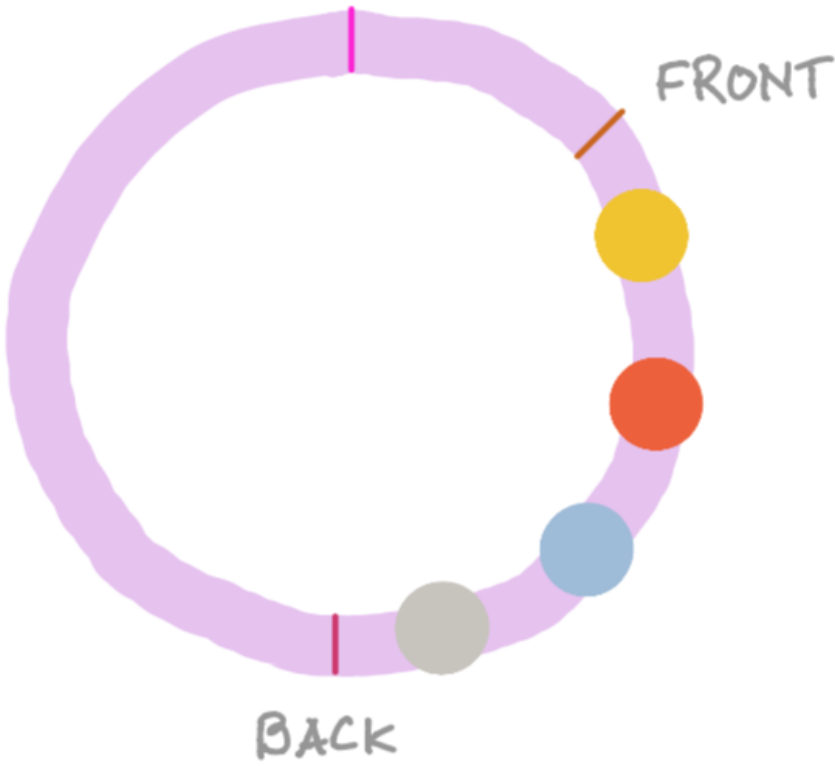
Popping/dequeueing an element in a linked list implementation of a **queue** or **stack**

Arrays

When implementing queues and stacks with arrays, it's important to visualize them as "circular" arrays. The beginning and end of an array do not matter to a stack or a queue. Simply keep track of the indices that locate the front and back of the queue/stack. One of the limitations with this implementation is that it gives a limit to the amount of elements the data structure can store.



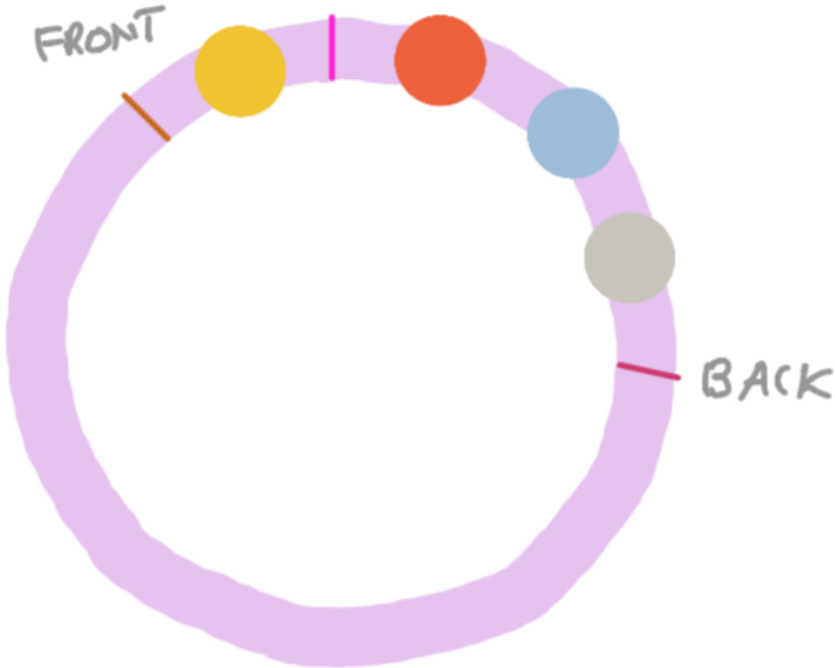
An array implementation of a queue/stack



A circular array visualization of a queue/stack



The queue/stack wraps around the ends of the array



A circular array visualization of a queue/stack