

Stack and Queue

Introduction

By: Zahra Bukhari

What are Stacks?

Stacks are a linear data structure that insert and remove items according to the **Last In First out**, or **LIFO**. Think of it as a kind of a fancy linked list. You will have to know array implementation for Rizk's exam.

Commonly used stack class functions

- **push(data)** Inserts element data to the top
- **pop()** Removes the top element
- **top()** Returns the top element
- **size()** Returns the number of elements
- **empty()** Returns true if empty

It is helpful to come up with your own analogy to understand stacks. When I took DS and was teaching myself this topic for an exam, I compared it to dirty dishes.

Let's imagine a scenario where Spongebob is closing up the Krusty Krab.



Spongebob is cleaning up tables. He is picking up the dishes from each table and stacking them on top of each other, one by one.

The last dish spongebob picks up will be all the way on the top of his stack, and the first dish will be all the way at the bottom. This process is similar to a stack **push** function. Spongebob is using **push(dish)**.



Now that he's done gathering his stack of dishes, Spongebob is going to wash each dish, one by one.

As we established before, the very last dish will be on top. So when Spongebob picks up the dish on the pile of his stack, the **last** dish **in** will be the **first** dish **out**.

This process is similar to a stack **pop()** function.



What are Queues?

Queues are a linear data structure that insert and remove items according to the **First In First out**, or **FIFO**. Think of it as a kind of a fancy linked list. You will have to know array implementation for Rizk's exam.

Commonly used queue class functions

- **push(data), or enqueue(data)** Inserts element data to the back
- **pop(), or dequeue()** Removes the front element
- **front()** Returns the front element
- **size()** Returns the number of elements
- **empty()** Returns true if empty

Let's think of this as a line, or a *queue*, in a restaurant.

Let's imagine a scenario where the Krusty Krab receives the sardine lunch rush.



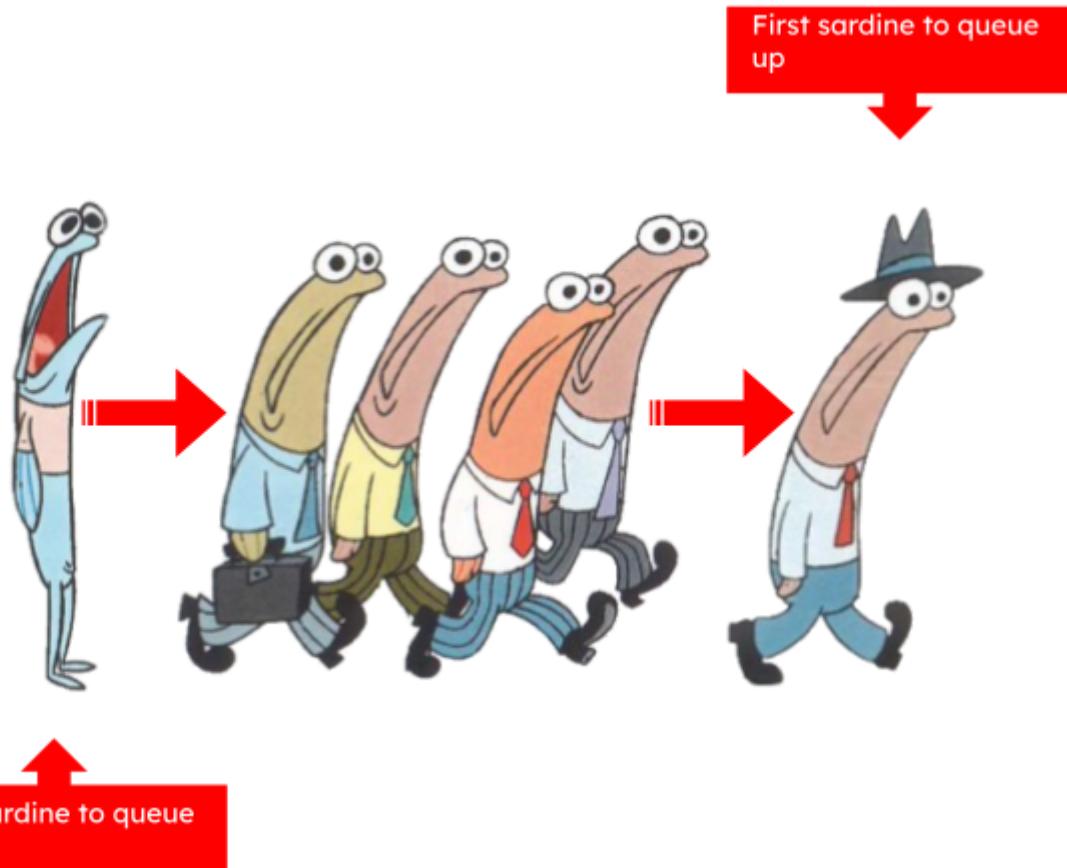
The sardines rush in and crowd Squidward and Mr. Krabs.



Squidward screams at the sardines to form a line, which will be our queue.



The sardines line up one by one, the first sardine is at the front of the line and the last sardine is at the back. This process is similar to a queue **enqueue** function. Squidward is using **enqueue(sardine)**.



When Squidward starts to take each sardine's order one by one, the **first** sardine **in** will be the **first** sardine **out**. Then the sardine behind him becomes the sardine in the front position.

This process is similar to a queue **dequeue()** function.

