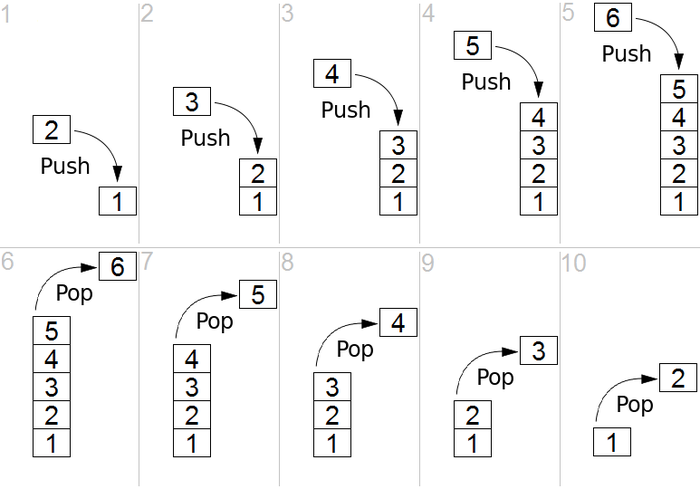
**What are stacks?**



A stack is a data structure with two main operations: **push** and **pop.**

**Push:** append an element on top of the stack

**Pop:** remove an element from the top of the stack

Think of a tray holder in a dining hall as a real world example of what a stack is. You can take out one tray from the top of the holder, and you can put a different one on top of the holder. The interesting fact with a stack is that you can only operate on the data on top of it.

For this reason, stacks are referred to as a **LIFO** or a **FILO** data structure.

LIFO = Last in First out

FILO = First in Last out

The two terms points to the same characteristic of the stack: if you push an element to the stack, you can remove it only if you remove the other elements that are added to the stack after itself (think about removing the element 1 from the stack).

### When are stacks useful?

Stacks are useful in two ways:

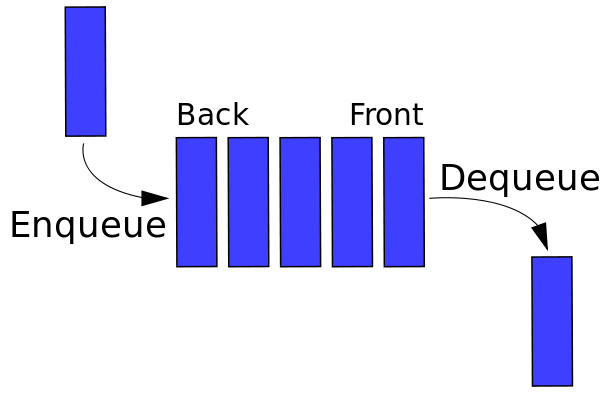
**1. Tracing back to access previous elements/operations**

For example, undo operations in editors are like popping a code change that was just pushed in the stack of edit history. Similarly, back operations in browsers are like popping a site visit that was just pushed in the stack of browser history.

**2.** **Matching recursive elements/operations**

For example, a stack comes in handy when checking if an operation “(5 + 6) + ((7 \* 9) + 9) + 10)” is a valid syntax (see sample problem 3 for details). Another use case is to keep track of recursions in a code. Each call to the recursive function is pushed onto the stack so that it can be executed once lower recursions are done with their execution.

### ****What are queues?****



A queue is a data structure with two main operations: **enqueue** and **dequeue**.

**enqueue:** append an element to the tail of the queue

**dequeue:** remove an element from the head of the queue

Queues should be easier to understand, since queues in computer science are just like queues in real life. Think of a long line in front of a busy restaurant. Each person will be “enqueued” to the line and once they reach the head of the line, they are “dequeued” and enters the restaurant.

Whereas stacks are to as LIFO or FILO, queues are referred to as a **FIFO** or **LILO** data structure.

FIFO = First in First out

LILO = Last in Last out

The two terms points to the same characteristic of the queue: the first person to join the queue is the first person to get out of the queue to enter the restaurant. The last person to join the queue is the last person to enter the restaurant.

### When are queues useful?

Queues are used whenever you want to process things one at a time as they come in.

Some examples are, uploading bunch of images, printing multiple documents, and processing thousands of requests to a web server.