Stacks

Definition 0.1

A **stack** is a last in, first out (LIFO) linear data structure, meaning that additions and removals happen on the same side of the structure.

The main operations for stacks include:
- **push(data)** - adds the data to the "top" of the stack
- **pop()** - removes the data at the top of the stack and returns it
- **peek()** - returns data for the top of the list without removing

SLL-Based Stack

- Does not need a tail pointer

Note 0.1

An SLL based stack uses the *front of the SLL as the top of the stack*. Thus, push simply becomes addToFront and pop becomes removeFromFront, both of which are **O(1) operations**

Array-Based Stack

- Requires a size variable along with the array

Note 0.2

In this case, the top of the stack is the back of the array. So we push by adding data to **arr[size]** and pop by removing the value at **arr[size-1]**, both of which are **O(1)** operations.