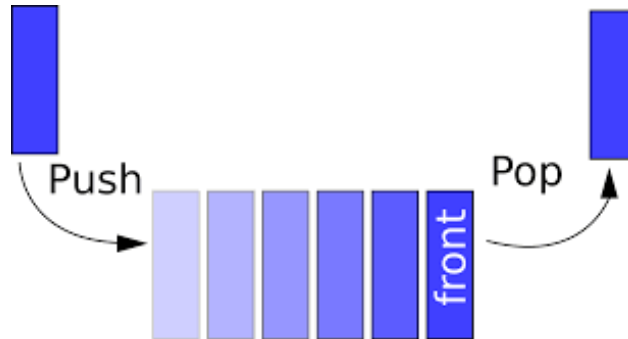


## The Queue data structure

- Stack: Last in, first out (LIFO). You only have access to the top item.
- Queue: First in, first out (FIFO). There are two ends, the back and the front. New items are added to the back and are taken from the front.



Applications of queues:

- To-do list.
- Queue of instructions to be executed by the CPU.
- Queue of messages to be sent by an Internet server.

## LinkedList

Because a Queue has two ends, it becomes necessary to use “double links.” A double link has references to both the next link and the previous link. Note that the Link class in the LinkedList project now includes a previous instance variable. Every link in a queue should be linked both ways, with two exceptions:

- The back link has no “previous” link.
- The front link has no “next” link.

## ArrayQueue

Similar to lists and stacks, a Queue can be implemented using an array. With an array, the “front” of a queue is located at lower indices, and the “back” of the queue is at higher indices. There are two int variables that store the locations of the back and front.

**push() and pop() example with an array.**

new APLinkedQueue()  
size = 0

back									
front									

push(A)  
size = 1

A									
front	back								

push(B)  
size = 2

A	B								
front		back							

push(C)  
size = 3

A	B	C							
front			back						

pop() returns A  
size = 2

	B	C							
	front		back						

pop() returns B  
size = 1

		C							
		front	back						

... and so on.

## Cycling the items

From the examples above, you can see that the items tend to “march” to the right as the queue is filled and emptied. So you might get to the end of the array even if the queue is not full. To avoid resizing, you should wrap around the array if possible. Only resize the array when there are actually too many items.

For example, if we do:

```
APArrayQueue<Integer> q = new APArrayQueue<Integer>();  
for(int i = 0; i < 10; i++) {  
    q.push(i);  
}  
  
q.pop()    // Returns 0  
q.pop()    // Returns 1  
q.pop()    // Returns 2  
  
q.push(10);  
q.push(11);
```

The state of the array should be,

10	11		3	4	5	6	7	8	9
		back	front						

In this example, the “back” variable wrapped around the end.