Lists

A list is an ordered collection of items. By ordered, we mean that it has a 1st, 2nd, ith, last, etc.

Because lists are ordered, we usually refer to items by position rather than by name.

Also, duplicate items are okay because we can differentiate by position.

Two common examples of lists are stacks and queues.

A stack is a list that obeys last in first out (LIFO) order. The last item added is always the first item removed

Special names for operations

Add = push O(1)

Remove = pop O(1)

Assess = top O(1)

Let’s implement a stack using an array, we can think of a stack as a list in which insertions and deletions happen at the same end. We will obviously choose the back end of our array.

A queue is a list that obeys first in first out order. The first item added is always the first item removed

Special names for the operations

Add = enqueue

Remove = dequeue

Assess = onqueue

To implement a queue using an array, we can think of the operations happening on opposite ends.

Attempt #1: insert at front, delete from back

The shifting: O(n) O(1)

Attempt #2: insert at back, delete from front

O(1) the shifting: O(n)

Let’s be clever! Let’s avoid shifting. We will keep track of the location holds the item (first one). We will also keep track of the number of items as usual

Let’s use modulo arithmetic to make this happen.

By using a circular buffer, we can support add and remove in O(1) time.