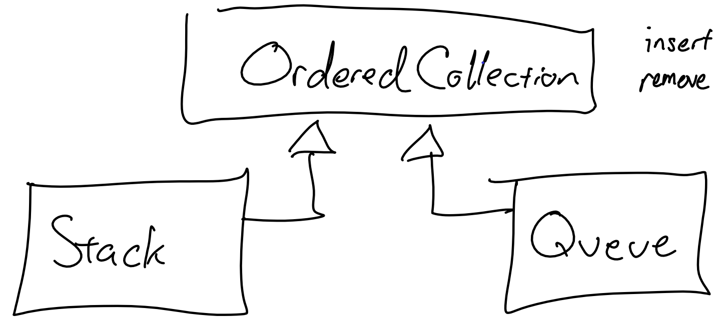
Stacks, Queues, Implementation



	LinkedList				ArrayList			
insert	insertFirst $O(1)$	insertLast $O(1)$	insertFirst $O(1)$	insertLast $O(1)$	insertFirst $O(n)$	insertLast amort. $O(1)$	insertFirst $O(n)$	insertLast amort. $O(1)$
remove	removeFirst $O(1)$	removeFirst $O(1)$	removeLast $O(n)$	removeLast $O(n)$	removeFirst $O(n)$	removeFirst $O(n)$	removeLast $O(1)$	removeLast $O(1)$
ADT Implemented	Stack	Queue	Queue	Stack	Stack	Queue	Queue	Stack

Induction

Correctness

```
Function isSorted(A, n):
    For i In 0...n-2:
        If A[i] > A[i+1]:
            Return False
        EndIf
    EndFor
    Return True
EndFunction
```

$O(n)$

```
Function isSorted(A, n):
    Return False
EndFunction
```

$O(1)$

$$\underbrace{1+\underbrace{2+3+\cdots+(n-2)+(n-1)}_{n+1}+n}_{n+1} = (n+1)\frac{n}{2} = \frac{n(n+1)}{2}$$

Proof by Mathematical Induction

1. State "proposition" — a statement which is either true or false
2. Prove base case (e.g. $P(1)$)
3. Prove inductive step:

   If $P(k)$ then $P(k+1)$

$$P(n) \equiv 1+2+\cdots+(n-1)+n = \frac{n(n+1)}{2}$$

$$\Downarrow$$

$$P(1) \equiv 1 = \frac{1(1+1)}{2}$$

$$P(2) \equiv 1+2 = \frac{2(2+1)}{2}$$

$$\vdots$$

So by 2: $P(1)$
So by 3: $P(2)$
So by 3: $P(3)$
$\vdots$

PMI: $\boxed{\sum_{i=1}^{n} i} = \frac{n(n+1)}{2}$

$1+2+\cdots+(n-1)+n$

Step 1. $P(n) \equiv \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Step 2. Want to show $P(1)$ is true.

WTS $\sum_{i=1}^{1} i = \frac{1(1+1)}{2}$

$$1 = \frac{1\cdot\cancel{2}}{\cancel{2}} \checkmark$$

Step 3. Assume $P(k)$ is true.
Want to show $P(k+1)$ is true.

WTS: $\sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$

Assume: $\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$

$$\left(\sum_{i=1}^{k} i\right)+(k+1) = \frac{k(k+1)}{2}+k+1$$

$$\sum_{i=1}^{k+1} i = \frac{k^2+k}{2}+k+1 = \frac{k^2+k+2k+2}{2} = \frac{k^2+3k+2}{2} = \frac{(k+1)(k+2)}{2}$$

$\checkmark$

Step 1

$$P(n) \equiv \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

$\boxed{\text{WTS } P(n) \text{ for all } n \geq 0}$

Reminder: $2^0 = 1$

Step 2

$$P(0) \equiv \sum_{i=0}^{0} 2^i = 2^{0+1} - 1$$

$$2^0 = 2^1 - 1$$
$$1 = 2 - 1 \checkmark$$

Step 3

If $P(k)$ then $P(k+1)$.

WTS $P(k+1)$.

$$\sum_{i=0}^{k+1} 2^i = 2^{k+2} - 1$$

Assume $P(k)$.

That is, assume $\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$.

$$\left(\sum_{i=0}^{k} 2^i\right) + 2^{k+1} = 2^{k+1} - 1 + 2^{k+1}$$

$$\sum_{i=0}^{k+1} 2^i = 2 \cdot 2^{k+1} - 1$$
$$= 2^{k+2} - 1$$

$\checkmark$

```
Function sumArray (A, n)
    If n=0 Then
        Return 0
    Else
        Return sumArray (A, n-1) + A[n-1]
    EndIf
EndFunction
```

Step 1.

$P(n) \equiv$ sumArray (A, n) returns the sum of the first n elements of the array A.

Step 2. Prove $P(0)$:

sumArray $(A, 0)$ returns the sum of the first 0 elements of the array A.

Proof. Because $n=0$, the condition of the If statement is true. So, we evaluate "Return 0", which returns 0, the sum of no numbers.

Step 3. Assume $P(k)$. Prove $P(k+1)$.

WTS: sumArray $(A, k+1)$ returns the sum of the first $k+1$ elements of A.

Because $k \geq 0$, $k+1 > 0$ and so the If condition is false. We call sumArray (A, k). By the inductive hypothesis, this returns the sum of the first k elements of A. We add $A[k]$ to this value, producing the sum of the first $k+1$ elements of A. This is what we return.