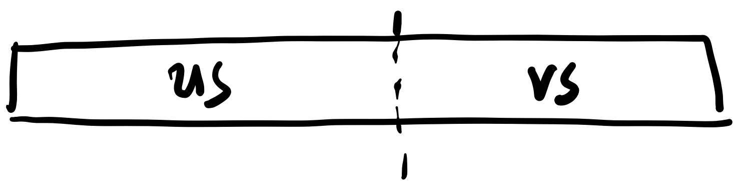
LECTURE 8 : Amortized Analysis

> "No man is an Iland,
> intire of itselfe;
> every man is a peece of the Continent,
> a part of the maine; "
>
> — John Donne, 1624
> Devotions upon Emergent Occasions.

Let's look at some list operations and their costs:

cons	$O(1)$	snoc	$O(n)$
head	$O(1)$	last	$O(n)$
tail	$O(1)$	init	$O(n)$

Let's restore some symmetry:

xs = 

Deque us sv where sv = reverse vs

```
> data Deque a = Deque [a] [a]
```

```haskell
> toList :: Deque a -> [a]
> toList (Deque us sv) = us ++ reverse sv
```

A deque is subject to two invariants:

$$\text{null us} \Rightarrow \text{null sv} \vee \text{single sv}$$
$$\text{null sv} \Rightarrow \text{null us} \vee \text{single us}$$

Now to make a Deque:

Naughty!

```haskell
fromList :: [a] -> Deque a
fromList xs = Deque xs []
```

O(n)

```haskell
> fromList :: [a] -> Deque a
> fromList xs = Deque us sv
>     where
>         (us, vs) = splitAt (n `div` 2) xs
>         n = length xs
>         sv = reverse vs
```

O(1)

```haskell
> cons :: a -> Deque a -> Deque a
> cons x (Deque us sv) = Deque (x:us) sv
```

similarly

```
> snoc :: Deque a -> a -> Deque a
> snoc (Deque [] sv) x = Deque sv [x]
> snoc (Deque us sv) x = Deque us (x:sv)
```

O(1)

```
> last :: Deque a -> a
> last (Deque us []) = head us -- or last us
> last (Deque us (v:sv)) = v
```

```
> tail :: Deque a -> Deque a
> tail (Deque [] sy) =
>        Deque [] (tail sy)                     O(1)
> tail (Deque [x] sy) =
>        Deque us sv                            O(n)
>     where ys = reverse sy
>           (us, vs) = splitAt (n `div` 2) ys
>           sv = reverse vs
>           n  = length ys.
> tail (Deque xs sy) = Deque (tail xs) sy
```

O(1)

In amortized complexity, we are concerned with a sequence of operations:

$$xs_0 \xrightarrow{op_0} xs_1 \xrightarrow{op_1} xs_2 \xrightarrow{op_2} \cdots \xrightarrow{op_n} xs_{n+1}$$

There are 3 different cost functions:

1. Real cost: $C_{op_i}(xs_i)$ [worst case]
2. Amortized cost: $A_{op_i}(xs_i)$
3. Size function: $S(xs)$

$$C_{op_i}(xs_i) \leqslant A_{op_i}(xs_i) + S(xs_i) - S(xs_{i+1})$$

$$\Rightarrow$$

$$\sum_{i=0}^{n+1} C_{op_i}(xs_i) \leqslant \sum_{i=0}^{n+1} A_{op_i}(xs_i) + S(xs_0) - S(xs_n)$$

$$\{\text{assume } S(xs_0) = 0\}$$

$$\Rightarrow \sum_{i=0}^{n+1} C_{op_i}(xs_i) \leqslant \sum_{i=0}^{n+1} A_{op_i}(xs_i)$$