Runtimes

```python
foo (arr):
    do
        swapped = false
        for i = 1 to len(arr)-1:
            if arr[i-1] > arr[i]:
                swap arr[i-1] and arr[i]
                swapped = true
    while swapped
```

What is the runtime?
- WC: \( \Theta(n^2) \) - all swaps
- BC: \( \Theta(n) \) - no swaps
- AC: \( \Theta(n^2) \) - half swaps

What is \( n \)?
- Array size
- Graph size (edges vs vertices)
- # of bits

Why do we care about runtime? Why not real execution time?

What about space?
- foo uses \( \Theta(n) \) space and \( \Theta(1) \) auxillary.

What counts as an atomic operation outside of bitland?
- Comparison
- Assignment
- Arithmetic operations (not exponentiation)

Note that we ignore memory management overhead. We're interested only in the complexity of operations on data structures.
Big O notation

We only care about asymptotic behavior.

\[ 2x \quad 5x + 4 \quad x + \sqrt{x} \]

These are all basically the same function eventually in performance.
We formalize this notion into equivalence classes.

Let \( f, g : \mathbb{N} \to \mathbb{N} \). We say \( f \in O(g) \) if \( \exists c \in \mathbb{R}^+ \forall n \geq n_0 \quad |f(n)| \leq c |g(n)| \)

**Ex:** Prove \( 2n^2 + n - 1 \in O(n^2) \).

\[
|2n^2 + n - 1| \leq 2n^2 + n^2 - 1 \quad (n \geq 1) \\
\leq 2n^2 + n^2 \quad \text{(everywhere)} \\
= 3n^2 \quad (n = 1) \\
= 3|n^2| \quad (n \geq 1)
\]

\[ c = 3 \quad N = 1 \]

**Ex:** Prove for \( f, g : \mathbb{N} \to \mathbb{N} \) that if \( f \in O(g) \), \( f + g \in O(g) \).

Since \( f \in O(g) \), \( \exists c_0 \in \mathbb{R}^+ \forall n \in \mathbb{N} : |f(n)| \leq c_0 |g(n)| \) when \( n \geq n_0 \).

Then when \( n \geq n_0 \),

\[
|f(n) + g(n)| \leq |f(n)| + |g(n)| \\
\leq c_0 |g(n)| + |g(n)| \\
= (c_0 + 1) |g(n)|.
\]

\[ c = c_0 + 1 \quad N = n_0 \]

**Obs:** Smaller terms don’t matter asymptotically.
Ex1 Prove $n^2 \neq O(n)$.
Let $c \in \mathbb{R}^+$ and $N \in \mathbb{N}$ be given.
Then when $n \geq \max(c, N) + 1 > 0$

$$|n^2| = n^2 \quad > cN \quad = c|N|,$$

hence $n^2 \notin O(n)$.

Other Classifications

**Big $O$**

$f \in O(g) \iff f \text{ is at most } g$.

**Little $o$**

$f \in o(g) \iff \forall c \in \mathbb{R}^+ \forall n, |f(n)| < c \lg(n)$

$f \in o(g) \iff f \text{ is smaller than } g$.

**Big $\Omega$**

$f \in \Omega(g) \iff \exists c \in \mathbb{R}^+ \forall n, |f(n)| \geq c \lg(n)$

$f \in \Omega(g) \iff f \text{ is at least } g$.

**Little $\omega$**

$f \in \omega(g) \iff \forall c \in \mathbb{R}^+ \forall n, |f(n)| \geq c \lg(n)$

$f \in \omega(g) \iff f \text{ is bigger than } g$.

**Thet$\vartheta$**

$f \in \Theta(g) \iff f \in O(g) \text{ and } f \in \Omega(g)$

$f \in \Theta(g) \iff f \text{ is the same as } g$. 
Complexity Relations

\[ f \in o(g) \Rightarrow f \in O(g) \]
\[ f \in \omega(g) \Rightarrow f \in \Omega(g) \]
\[ f \in O(g) \Rightarrow g \in \Omega(f) \]
\[ f \in o(g) \Rightarrow g \in \omega(f) \]

Limit Definitions

Consider the limit

\[ \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| = m \]

assuming the limit exists.

If the limit exists and \( m = 0 \), \( f \in o(g) \).

If the limit exists and \( m > 0 \), \( f \in \Theta(g) \).

Where does \( O \) fit into this?

If we flip \( f \) and \( g \) in \( \infty \), we get the same results for \( \omega, \Omega \).

How do we get results for \( \omega, \Omega \)?

(See complexity relations above.)