Other Complexity Measures

Big O and Friends

So far we have looked at Big O as a way to identify the complexity of an algorithm, and that is what we will be most concerned with. But there are others:

- **Big O**: $f(n) = O(g(n))$: g is an upper bound on how fast f grows as n increases.
- **Little o**: $f(n) = o(g(n))$: A stricter upper bound than Big O.
- **Theta**: $f(n) = \Theta(g(n)$: More precise than Big O and Little o, it provides both upper and lower bounds, which are given by the same function, except with different constant factors. That is, f and g grow at the same rate.
- **Asymptotically Equal**: $f(n) \sim g(n)$: stricter upper and lowerbounds
- **Omega**: $f(n) = \Omega(g(n))$: a lower bound on how fast f grows as n increases (the lower bound equivalent of big O)

Big O revisited

$$f(n) = O(g(n)) \iff |f(n)| \leq |Cg(n)|$$

for some constants C, n_0 where $n > n_0$

- f grows at the same rate or slower than g.
- But $2n^2 + n = O(n^2)$, so we can have $f(n) > g(n)$ for all n.
- So Big O only refers to relative growth rate, NOT relative speed or memory usage.

Little o

$$f(n) = o(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} \text{ exists and is equal to } 0$$

This makes g an upperbound on f but a stronger one than Big O:

Note that $2n^2 = O(n^3)$ and $2n^2 = O(n^2)$ (choose C = 3)

$2n^2 = o(n^3)$ because:

$$\lim_{n\to\infty} \frac{2n^2}{n^3} = \lim_{n\to\infty} \frac{2}{n}$$
$$= 0$$

But it is not true that $2n^2 = o(n^2)$ because:

$$\lim_{n\to\infty} \frac{2n^2}{n^2} = \lim_{n\to\infty} 2$$
$$= 2$$

Theta

$$f(n) = \Theta(g(n)) \iff c_1 g(n) \leq f(n) \leq c_2 g(n)$$

for positive constants c_1, c_2, n_0, and $n > n_0$

This means that f and g have the same rates of growth, with some constant multiple, i.e. that f is bounded above and below by (possibly different) multiples of g.

This is only true if $f(n) = O(g(n))$ and $g(n) = O(f(n))$

Example: $x^2 + 2x + 1 = \Theta(x^2)$

But it is not true that $x^2 + 2x + 1 = \Theta(x^3)$

Asymptotically Equal

$$f(n) \sim g(n) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} \text{ exists and is equal to 1}$$

This has the same relation to Theta that Little o has to Big O: Asymptotically Equal is a tighter upper and lowerbound than Theta.

$x^2 + x = \Theta(x^2)$ and $x^2 + x \sim x^2$

However, $2x^2 + x = \Theta(x^2)$ and it is **NOT** true that $2x^2 + x \sim x^2$

Omega

$$f(n) = \Omega(g(n)) \iff |f(n)| \geq |cg(n)|$$

for positive constants c, n_0 where $n > n_0$

This provides a lower bound on f: As f grows, it will always grow at least at the same rate as g and it could grow faster.