CIS 575. Introduction to Algorithm Analysis
Material for January 24, 2024

Little o, Little omega

The topic of this note in covered in *Cormen*'s Section 3.2.

1 Vastly Different Speeds of Growth

We have introduced Big O to denote "does not grow faster than". To denote "grows arbitrary slower than", we shall introduce *little o*: we say $f \in o(g)$ iff for **all real** $c > 0$ there exists n_0 such that for all n with $n \geq n_0$, $f(n) < cg(n)$.

> Observe that whereas the definition of big O demands that there *exists* a $c > 0$ such that $f(n) < cg(n)$ when n is beyond a certain threshold, the definition of little o demands that for *all* $c > 0$ it will eventually hold that $f(n) < cg(n)$, and in particular requires us to consider real values for c (as otherwise the property would just force f to be smaller than g, not arbitrarily smaller).

The property denoted by $f \in o(g)$ is well-known from calculus (where we would typically use "ε" instead of "c"); we see that

$$f \in o(g) \text{ if and only if } \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Just as Big O has a dual, Big Omega, also "little o" has a dual, *little omega* defined by:

$$g \in \omega(f) \text{ holds exactly when } f \in o(g).$$

With X ranging over the various symbols we have introduced, and R ranging over the various comparison operators, we have

$$n^p \in X(n^q) \text{ iff } p\ R\ q$$

where X corresponds to R as given by the table

X	R
o	$<$
O	$\leq$
Θ	$=$
Ω	$\geq$
ω	$>$

In general, it is easy to see that

- **if** $f \in o(g)$

- **then** $f \in O(g)$ **and** $f \notin \Theta(g)$

and the above table shows that the converse also holds when f and g are polynomials, but in general the converse does not hold as can be demonstrated by a counterexample where $g(n) = n$ whereas f is a rather contrived function:

$$f(n) = 2^{2^{\lfloor \lg \lg n \rfloor}}$$

Polynomials Good, Exponentials Bad From calculus, we know that any polynomial is eventually dominated by any exponential:

$$\text{for all } a > 0, \text{ and all } b > 1, \mathbf{n^a} \in \mathbf{o}(\mathbf{b^n})$$

In particular, as we mentioned in the first note for this class (while also discussing some philosophical implications),

$$n^{10000000} \in o((1.000000000001)^n)$$

This result partly justifies the maxim which underlies much of algorithm analysis:

- an algorithm that runs in polynomial time is good (or at least not too bad)
- while an algorithm that runs in exponential time is bad.

Dually, the *inverse* of any exponential is eventually dominated by the inverse of any polynomial, in particular by any root:

$$\text{for all } a > 0, \text{ and all } b > 1, \log_{\mathbf{b}}(\mathbf{n}) \in \mathbf{o}(\mathbf{n^a}) \text{ and } \log_{\mathbf{b}}(\mathbf{n}) \in \mathbf{o}(\sqrt[a]{n}).$$