Chapter 14-Big-O

Saturday, January 7, 2023
10:35 PM

Program Runtime and Asymptotic Relationships:

To determine the efficiency of a program, we need to model the runtime of the program. However, we are lazy, so we just chuck a big value at it and see its behavior.

To compare 2 function's efficiency, we take the ratio of the 2 functions.

Asymptotically similar: $f(n) \approx g(n)$ if and only if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$, where c is a real number.

Asymptotically smaller: $f(n) \ll g(n)$ if and only if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

(Note that not all functions can be evaluated this way, since some functions will oscillate and others just don't exist when put into a ratio, for those there are other methods)

Primitive Functions Growth Rate:

Higher-order polynomials grow faster than lower-order ones:

$n^2 \ll n^5$

Exponentials grow faster than normal high order polynomials:

$n^k \ll 2^n$

The base of the exponent also matters:

$2^n \ll 3^n$

A tricky one that's not too obvious:

$2^n \ll n!$

(since they both have n terms, but n! have bigger terms multiplied together)

Slow growing functions are better for computer programs.
(The log function, for example)

Algorithms for sorting a list of numbers have running times that grow like $n \log n$. We know from the above that $1 \ll \log n \ll n$. We can multiply this equation by n because factors common to f and g cancel out in our definition of $\ll$. So we have $n \ll n \log n \ll n^2$.

Dominant Term Method:

Most of the time we can just manipulate relationships at a high level and look at the primitive functions (the dominant expressions), and ignore the parts that grow slower.

(Note that a constant multiple difference matters)

Big- *O*:

For messy functions, we use big O to compare their running times.
Instead of a simple ratio, we say that a function f(n) is ***O(***g(n)***)*** iff

There are positive real numbers c and k such that $0 \leq f(n) \leq cg(n)$ for every $n \geq k$.

In most cases, g(n) is a simple reference function, and this helps us understand how slow/fast the function in question f(n) runs.

(Note that the function f can have the same/similar dominant term, regardless of a constant multiple difference, and the big o relationship still holds)

The Equivalence Relationship:

When $g(n)$ is $O(f(n))$ and $f(n)$ is $O(g(n))$, then $f(n)$ and $g(n)$ are forced to remain close together as n goes to infinity. In this case, we say that $f(n)$ is $\Theta(g(n))$ (and also $g(n)$ is $\Theta(f(n))$). The Θ relationship is an equivalence relation on this same set of functions. So, for example, the equivalence class $[n^2]$ contains functions such as n^2, $57n^2 - 301$, $2n^2 + n + 2$, and so forth.

Showing the Big-O Relationship:

To show that a big-O relationship holds, we just need to find an *n* and *k* that work for the 2 functions.
(Note that you can pick **any** *n* and *k,* as long as it works)

For example, to show that $3n$ is $O(n^2)$, we can pick $c = 3$ and $k = 1$. Then $3n \leq cn^2$ for every $n \geq k$ translates into $3n \leq 3n^2$ for every $n \geq 1$, which is clearly true. But we could have also picked $c = 100$ and $k = 100$.

(if you don't want to use overkill values every time, or if you don't know if a value is overkill enough, then pick a c value first, then plug it into the inequality. Then you can simplify the equation and start plugging and chugging)

Proving a Primitive Function Relationship:

Claim 50 *For every positive integer $n \geq 4$, $\frac{2^n}{n!} < (\frac{1}{2})^{n-4}$*

Proof: Suppose that n is an integer and $n \geq 4$. We'll prove that $\frac{2^n}{n!} < (\frac{1}{2})^{n-4}$ using induction on n.

Base: $n = 4$. [show that the formula works for $n = 4$]

Induction: Suppose that $\frac{2^n}{n!} < (\frac{1}{2})^{n-4}$ holds for $n = 4, 5, \ldots, k$. And, in particular, $\frac{2^k}{k!} < (\frac{1}{2})^{k-4}$

We need to show that the claim holds for $n = k + 1$, i.e. that $\frac{2^{k+1}}{(k+1)!} < (\frac{1}{2})^{k-3}$

(Note that when it comes to proving inequalities, you should write out explicitly the full inductive hypothesis and conclusion, like how both the assumed inequality and inequality to be proved are written out above)

Induction: Suppose that $\frac{2^n}{n!} < (\frac{1}{2})^{n-4}$ holds for $n = 4, 5, \ldots, k$.

Since $k \geq 4$, $\frac{2}{k+1} < \frac{1}{2}$. So $\frac{2^{k+1}}{(k+1)!} < \frac{1}{2} \cdot \frac{2^k}{k!}$.

By our inductive hypothesis $\frac{2^k}{k!} < (\frac{1}{2})^{k-4}$. So $\frac{1}{2} \cdot \frac{2^k}{k!} < (\frac{1}{2})^{k-3}$.

So then we have $\frac{2^{k+1}}{(k+1)!} < \frac{1}{2} \cdot \frac{2^k}{k!} < (\frac{1}{2})^{k-3}$ which is what we needed to show.