CSE 2320 Notes 2: Growth of Functions

(Last updated 8/28/06 2:31 PM)

CLRS, Chapter 3

Why constants are annoying . . .

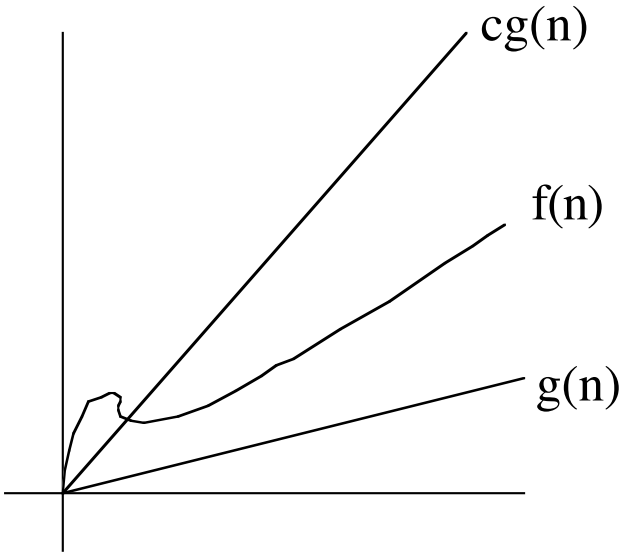
Algorithm
⇓
Language
⇓
Compiler
⇓
Machine

Need to compare time and space usage of various algorithms for a problem.

ASYMPTOTIC NOTATION

$O(g(n))$ is a *set* of functions:

$f(n) \in O(g(n))$ iff $\exists$ c and n_0 such that $f(n) \le cg(n)$ when $n \ge n_0$



Theorem: If $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ is a constant, then $f(n) \in O(g(n))$.

$f(n) = n^2, g(n) = n^3$ $\lim_{n\to\infty} \frac{n^2}{n^3} = \lim_{n\to\infty} \frac{2n}{3n^2} = \lim_{n\to\infty} \frac{2}{6n} = 0 \Rightarrow n^2 \in O(n^3)$

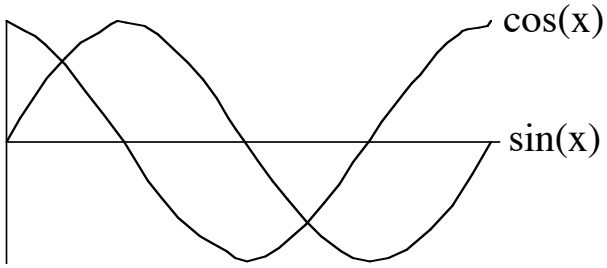
$f(n) = n^3,\ g(n) = n^2 \quad \lim_{n\to\infty} \frac{n^3}{n^2} = \lim_{n\to\infty} \frac{3n^2}{2n} = \lim_{n\to\infty} \frac{6n}{2} = \textit{unbounded} \Rightarrow n^3 \notin \mathrm{O}(n^2)$

$f(n) = 3n^2 + 2n - 3,\ g(n) = 5n^2 - n + 2$

$$\lim_{n\to\infty} \frac{3n^2 + 2n - 3}{5n^2 - n + 2} = \lim_{n\to\infty} \frac{6n + 2}{10n - 1} = \lim_{n\to\infty} \frac{6}{10} = \frac{3}{5} \Rightarrow 3n^2 + 2n - 3 \in \mathrm{O}(5n^2 - n + 2)$$

Conclusion: Toss out low-order terms $n^k \in \mathrm{O}\left(n^l\right)$ if $l \geq k$.

Is $\sin(x) \in \mathrm{O}(\cos(x))$?



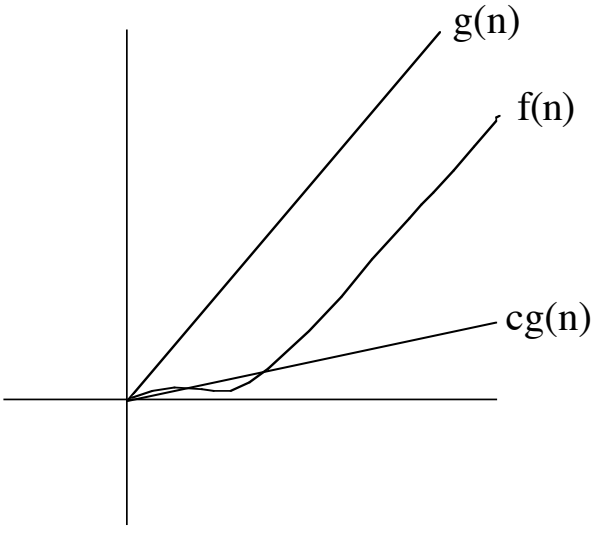
Is $\ln(x) \in \mathrm{O}(\log_{10}(x))$?

Is $2^n \in \mathrm{O}\left(n^k\right)$ for some *fixed k*?

$n^k \in \mathrm{O}\left(2^n\right)$ for any *k*. General case, $n^k \in \mathrm{O}\left(c^n\right)$ for any $c > 1$

$\Omega(g(n))$ is a *set* of functions:

$f(n) \in \Omega(g(n))$ iff $\exists\ c$ and n_0 such that $f(n) \geq cg(n) \quad (c > 0)$ when $n \geq n_0$



Theorem: If $\lim_{n\to\infty} \frac{g(n)}{f(n)}$ is a constant, then $f(n) \in \Omega(g(n))$.

$g(n) = n^2,\ f(n) = 3n^3\ \lim_{n\to\infty} \frac{n^2}{3n^3} = \lim_{n\to\infty} \frac{2n}{9n^2} = \lim_{n\to\infty} \frac{2}{18n} = 0 \Rightarrow n^3 \in \Omega(n^2)$

Upper bounds (O()) are for particular *algorithms*.

Lower bounds (Ω()) are for indicating the inherent difficulty of *problems*.

The hunt for an optimal algorithm . . .

Θ-notation (asymptotically *tight* bound) – Used for worst-case for algorithm

$f(n) \in \Theta(g(n))$ iff $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.

Theorem: If $\lim_{n\to\infty} \frac{g(n)}{f(n)}$ is a constant > 0, then $f(n) \in \Theta(g(n))$.

$f(n) = 3n^2 + 2n - 3,\ g(n) = 5n^2 + n - 1$

$$\lim_{n\to\infty} \frac{3n^2 + 2n - 3}{5n^2 + n - 1} = \lim_{n\to\infty} \frac{6n + 2}{10n + 1} = \lim_{n\to\infty} \frac{6}{10} = \frac{3}{5} \Rightarrow 3n^2 + 2n - 3 \in \Theta(5n^2 + n - 1)$$

$\Theta(f(n))$ is an equivalence relation . . .

Symmetric: $g(n) \in \Theta(f(n)) \Leftrightarrow f(n) \in \Theta(g(n))$ [Doesn't work for O or Ω]

Reflexive: $f(n) \in \Theta(f(n))$ [Works for O and Ω]

Transitive: $f(n) \in \Theta(g(n))$ and $g(n) \in \Theta(h(n)) \Rightarrow f(n) \in \Theta(h(n))$ [Works for O and Ω]

Accepted notational abuses of asymptotic notation.

1. $n^2 + n = \Theta\left(n^2\right)$ means $n^2 + n \in \Theta\left(n^2\right)$.

2. $3n^2 + 2n + 1 = 3n^2 + \Theta(n)$

3. $2 + 3\Theta(n) = n + \Theta(n)$

$n^3 + 2n^2 + 1 = n^3 + \Theta\left(n^2\right)$

Aside: Indicating that a bound could be improved:

$o(f) = O(f) - \Theta(f)$ $\quad$ $\omega(f) = \Omega(f) - \Theta(f)$

$n^2 \in o(n^3)$ $\quad$ $\frac{1}{n^2} \in \omega\left(\frac{1}{n^3}\right)$

CLRS 3.1-1 $f(n)$ and $g(n)$ are positive functions. Show $\max(f(n),g(n)) = \Theta(f(n) + g(n))$

O: Since $f(n) \le f(n) + g(n)$ and $g(n) \le f(n) + g(n)$

$\max(f(n),g(n)) \le f(n) + g(n)$ for all n, so $n_0 = 0$ and $c = 1$

Ω: Must have $c(f(n) + g(n)) \le \max(f(n),g(n))$, so choose $n_0 = 0$ and $c \le 1/2$ [Consider $f(n) = g(n)$]

Test Problem: Prove that if $g(n) \in \Omega(f(n))$ then $f(n) \in O(g(n))$.

$cf(n) \le g(n)$ when $n \ge n_0$
So, $f(n) \le \frac{1}{c} g(n)$ when $n \ge n_0$
Need $f(n) \le dg(n)$ when $n \ge n_1$
Choose $d = \frac{1}{c}$ and $n_1 = n_0$

Test Problem: Prove that if $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$, then $f(n) \in O(h(n))$.

$f(n) \le cg(n)$ when $n \ge n_0$ and $g(n) \le dh(n)$ when $n \ge n_1$
So, $cg(n) \le cdh(n)$ when $n \ge n_1$
Also, $f(n) \le cdh(n)$ when $n \ge \max(n_0,n_1)$.
Need $f(n) \le eh(n)$ when $n \ge n_2$.
Choose $e = cd$ and $n_2 = \max(n_0,n_1)$.

Time for the following code (`byRows()`):

```
for (i=0; i<n; i++)
  for (j=0; j<n; j++)
  {
    c[i][j] = 0;
    for (k=0; k<n; k++)
      c[i][j] += a[i][k]*b[k][j];
  }
```

Time for the following code (`byColumns1()`):

```
for (j=0; j<n; j++)
  for (i=0; i<n; i++)
  {
    c[i][j] = 0;
    for (k=0; k<n; k++)
      c[i][j] += a[i][k]*b[k][j];
  }
}
```

Time for the following code (`byColumns2()`):

```
int bColumn[n];

for (j=0; j<n; j++)
{
  for (k=0; k<n; k++)
    bColumn[k] = b[k][j];
  for (i=0; i<n; i++)
  {
    c[i][j] = 0;
    for (k=0; k<n; k++)
      c[i][j] += a[i][k]*bColumn[k];
  }
}
}
```

```
omega.uta.edu> cc matmult.c
omega.uta.edu> a.out
byRows() used 47.209999 CPU
byColumns1() used 45.140003 CPU
byColumns2() used 33.739998 CPU

omega.uta.edu> cc -O1 matmult.c
omega.uta.edu> a.out
byRows() used 18.239999 CPU
byColumns1() used 23.549999 CPU
byColumns2() used 16.970001 CPU
```

```
omega.uta.edu> cc -O2 matmult.c
omega.uta.edu> a.out
byRows() used 19.220001 CPU
byColumns1() used 22.049997 CPU
byColumns2() used 17.150002 CPU

omega.uta.edu> cc -O3 matmult.c
omega.uta.edu> a.out
byRows() used 19.480001 CPU
byColumns1() used 20.980000 CPU
byColumns2() used 17.000000 CPU
```

Time for the following code?

```
for (i=0; i<k; i++)
  ;
```

floor(x) = $\lfloor x \rfloor$ Largest integer $\le x$ $\lfloor 2.5 \rfloor = ?$ $\lfloor -2.5 \rfloor = ?$

ceiling(x) = $\lceil x \rceil$ Smallest integer $\ge x$ $\lceil 2.5 \rceil = ?$ $\lceil -2.5 \rceil = ?$

Worst-case number of probes for binary search: $\lfloor \lg n \rfloor + 1$

Consider worst-case searches on tables with 15 elements (4) and 16 elements (5).

Quick Review: $2^n \,?\, O(3^n)$ $2^n \,?\, \Omega(3^n)$ $\lg(2^n) = n$ $\lg(3^n) = n \lg 3$ $\lg(2^n) = \Theta(\lg(3^n))$

Factorials:

$$n! = \prod_{i=1}^{n} i = 1 \bullet 2 \bullet 3 \bullet \ldots \bullet n$$

$$\lg n! \in \Theta(n \lg n)$$

O: $n! \le n^n$

$$\lg(n!) \le n \lg n$$

$$\lg(n!) \in O(n \lg n)$$

Ω: $n! = 1 \bullet \ldots \bullet \frac{n}{2} \bullet \left(\frac{n}{2} + 1\right) \bullet \ldots \bullet n$ $\frac{n}{2} + k \ge \sqrt{n}$ for non-negative k with $n > 3$

$$n! \ge (\sqrt{n})^{\frac{n}{2}}$$

$$\lg(n!) \ge \frac{n}{2} \frac{\lg n}{2} = \frac{1}{4} n \lg n$$