Analysis of Algorithms

Growth of Functions

• Growth of Functions
• Asymptotic Notation: $O$, $\Omega$, $\Theta$, $o$, $\omega$
• Asymptotic Notation Properties

Growth of Functions

\[
\begin{align*}
n & \quad n \\
0.5n + 3 & \quad 0.5n + 3 \\
2n - 7 & \quad 2n - 7
\end{align*}
\]

Linear

Growth Rates

\[
\begin{align*}
2^n & \quad 2^n \\
n^2 & \quad n^2 \\
\log n & \quad \log n \\
n & \quad n
\end{align*}
\]
Growth Rates

\[ \lim_{n \to \infty} \frac{f(n)}{g(n)} = \begin{cases} 
0 & \text{if } f(n) \text{ grows slower than } g(n) \\
\infty & \text{if } f(n) \text{ grows faster than } g(n) \\
\text{otherwise} & \text{if } f(n) \text{ and } g(n) \text{ have the same growth rate}
\end{cases} \]

\[ f(n) \prec g(n) : f(n) \text{ grows slower than } g(n) \]

\[ 0.5^n < 1 < \log n < \log^6 n < n^{0.5} < n^3 < 2^n < n! \]

l'Hôpital's Rule

If \( f(n) \) and \( g(n) \) are differentiable, \( \lim_{n \to \infty} f(n) = \infty, \)

\[ \lim_{n \to \infty} g(n) = \infty, \text{ and } \lim_{n \to \infty} \frac{f'(n)}{g'(n)} \text{ exists, then} \]

\[ \lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{f'(n)}{g'(n)} \]

\[ f(n) = \log n \quad g(n) = \sqrt{n} \]

\[ \lim_{n \to \infty} \frac{\log n}{\sqrt{n}} = \lim_{n \to \infty} \frac{\ln n}{\ln 2^{\sqrt{n}}} = \frac{1}{\ln 2} \lim_{n \to \infty} \frac{\ln n}{\sqrt{n}} = \frac{1}{\ln 2} \lim_{n \to \infty} \frac{\ln (\sqrt{n})}{\sqrt{n}} = \frac{1}{\ln 2} \lim_{n \to \infty} \frac{2}{\sqrt{n}} = 0 \]

\[ \frac{d \ln n}{d n} = \frac{1}{n} \]

\[ \frac{d \sqrt{n}}{d n} = \frac{1}{(2\sqrt{n})} \]

\[ \text{Every sublinear function grows faster than any polylogarithmic function} \]

\[ \text{e.g., } n^{0.01} \text{ vs. } (\log n)^{100} \]
Asymptotic

Asymptotic : any approximation value that gets closer and closer to the truth, when some parameter approaches a limiting value.

Asymptotic Notations

- Deal with the behaviour of functions in the limit (for sufficiently large value of its parameters)
- Permit substantial simplification (napkin mathematics, rough order of magnitude)
- Classify functions by their growth rates

“Same” Growth Rates

Θ (1) Θ (log n) Θ (n^{0.5}) Θ (n log n) Θ (n^5)

“No Faster Than” Growth Rates

O (1) O (log n) O (n^{0.5}) O (n log n) O (n^5)
Special Orders of Growth

- constant : \( \Theta(1) \)
- logarithmic : \( \Theta(\log n) \)
- polylogarithmic : \( \Theta(\log^c n), \ c \geq 1 \)
- sublinear : \( \Theta(n^a), \ 0 < a < 1 \)
- linear : \( \Theta(n) \)
- quadratic : \( \Theta(n^2) \)
- polynomial : \( \Theta(n^c), \ c \geq 1 \)
- exponential : \( \Theta(c^n), \ c > 1 \)
**Analogy**

- $f(n)$ and $g(n)$ (real numbers)
- $f(n) = \Theta(g(n)) \approx f = g$
- $f(n) = O(g(n)) \approx f \leq g$
- $f(n) = o(g(n)) \approx f < g$
- $f(n) = \Omega(g(n)) \approx f \geq g$
- $f(n) = \omega(g(n)) \approx f > g$

Not all functions are asymptotically comparable

$n$ vs. $n^{1+\sin n}$

---

**O-notation**

$O(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that }$

$0 \leq f(n) \leq c \cdot g(n) \text{ for all } n \geq n_0 \}$

$27n^2 + 0.5n \in O(n^3)$ ?

$27n^2 + 0.5n \leq \Theta(n^3)$ for all $n \geq 0$

---

**Asymptotic Upper Bound**

$f(n) = \Theta(g(n))$

---

**Ω-notation**

$\Omega(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that}$

$0 \leq c \cdot g(n) \leq f(n) \text{ for all } n \geq n_0 \}$

$27n^2 + 0.5n \in \Omega(n)$ ?

$\Omega(n) \leq 27n^2 + 0.5n \text{ for all } n \geq 0$
Asymptotic Lower Bound

\[ f(n) = \Omega(g(n)) \]

\[ n \]

\[ c \cdot g(n) \]

\[ n_0 \]

\[ f(n) = \Omega(g(n)) \]

\[ \Theta \text{-notation} \]

\[ \Theta(g(n)) = \{ f(n) : \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that} \]

\[ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0 \} \]

\[ 27n^2 + 0.5n \in \Theta(n^2) \]

\[ 27n^2 \leq 27n^2 + 0.5n \leq \Theta(n^2) \text{ for all } n \geq n_0 \]

Asymptotic Tight Bound

\[ f(n) = \Theta(g(n)) \]

\[ f(n) = o(g(n)) \]

\[ \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \]

\[ f(n) = \omega(g(n)) \]

\[ \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \]

O-notation and \( \omega \)-notation
Asymptotic Notation Properties

- Transitivity
- Reflexivity
- Symmetry
- Transpose symmetry

Transitivity

- $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ imply $f(n) = \Theta(h(n))$
- $f(n) = O(g(n))$ and $g(n) = O(h(n))$ imply $f(n) = O(h(n))$
- $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$ imply $f(n) = \Omega(h(n))$
- $f(n) = o(g(n))$ and $g(n) = o(h(n))$ imply $f(n) = o(h(n))$
- $f(n) = \omega(g(n))$ and $g(n) = \omega(h(n))$ imply $f(n) = \omega(h(n))$

Reflexivity

- $f(n) = \Theta(f(n))$
- $f(n) = O(f(n))$
- $f(n) = \Omega(f(n))$

Symmetry and Transpose Symmetry

- $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$
- $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$
- $f(n) = o(g(n))$ if and only if $g(n) = o(f(n))$
- $f(n) = \omega(g(n))$ if and only if $g(n) = \omega(f(n))$
\[ f(n) = \Theta(g(n)) \text{ if and only if} \]

- \[ f(n) = \Omega(g(n)) \]
- \[ f(n) = O(g(n)) \]

**Example**

\[
\sum_{i=1}^{n} i^k = \Theta(n^{k+1})
\]

| O | \[
\sum_{i=1}^{n} i^k \leq \sum_{i=1}^{n} n^k = n^{k+1} = O(n^{k+1})
\]

| Ω | \[
\sum_{i=1}^{n} i^k \geq \sum_{i=1}^{n} \left( \frac{n}{2} \right)^k \geq \frac{n}{2} \left( \frac{n}{2} \right)^k = \Omega(n^{k+1})
\]

**Logarithms**

- Asymptotically, in logarithm,
  - the base of the log does not matter

  \[
  \log_b n = \frac{\log_c n}{\log_c b}
  \]

  \[
  \log_{10} n = \frac{\log_2 n}{\log_2 10} = \Theta(\log n)
  \]

  - any polynomial function of \( n \) does not matter

  \[
  \log n^{30} = 30 \log n = \Theta(\log n)
  \]

- \(\lg, \ln, \log\) (CS) (math) (asymptotic)
**Polylog, Polynomial, Exponential**

- Any positive exponential function grows faster than any polynomial function
  \[ f(n) : \text{monotonically growing function}, \quad (f(n))^c = o(a^{f(n)}), \quad c > 0, \quad a > 1 \]
- Any positive polynomial function grows faster than any polylogarithmic function
  \[ \log^b n = o(n^c) \]

**One-Way Equality**

- \( n = O(n^2) \Rightarrow n \in O(n^2) \)
- \( O(n^2) = n \)

- Why don’t we use \( \in \)? (GKP)
  - tradition: the practice stuck
  - tradition: CS is used to abuse equal sign
  - tradition: read “=“ as “is”, “is” is one-way
  - natural: when we do asymptotic calculation

**Asymptotic Notation in Equations**

- \( H_n = 1/1 + 1/2 + \ldots + 1/n = \ln n + \gamma + O(1/n) \)
- \( 2n^2 + 3n + 1 = 2n^2 + \Theta(n) \)
- \( 2n^2 + 3n + 1 = 2n^2 + f(n), \quad f(n) = \Theta(n) \)

- Eliminate inessential detail
  e.g., MergeSort: \( T(n) = 2T(n/2) + n \quad ? \)
  \[ T(n) = 2T(n/2) + (n-1) \quad ? \]
  \[ T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n) \]

**Manipulating Asymptotic Notations**

- \( c \cdot O(f(n)) = O(f(n)) \)
- \( O(O(f(n))) = O(f(n)) \)
- \( O(f(n))O(g(n)) = O(f(n)g(n)) \)
- \( O(f(n)g(n)) = f(n)O(g(n)) \)
- \( O(f(n) + g(n)) = O(\max(f(n), g(n))) \)
- \( \sum_{k=1}^{n} O(f(k)) = O(\sum_{k=1}^{n} f(k)) \)
Example: BuildHeap

\[
\begin{align*}
    k &= \lfloor \log n \rfloor \\
    \sum_{h=0}^{k} \left( \frac{n}{2^h} O(h) \right) &= O \left( n \sum_{h=0}^{k} \frac{h}{2^h} \right) \\
    &= O \left( n \sum_{h=0}^{k} \frac{h}{2^h} \right) \\
    &= O(2n) \\
    &= O(n)
\end{align*}
\]

Conclusion

- Growth of functions
  - give a simple characterization of function’s behavior
  - allow us to compare the relative growth rates of functions

- Use asymptotic notation to classify functions by their growth rates

- Asymptotics is the art of knowing where to be sloppy and where to be precise

Don Knuth

- Father of “analysis of algorithm”
- Author of “The Art of Computer Programming”
- Programmer of the T\TeX and METAFONT
- ...