Algorithms and Data Structures
CS 372

Asymptotic notations

(Based on slides by M. Nicolescu)

Algorithm Analysis

- The amount of resources used by the algorithm
  - Space
  - Computational time
- Running time:
  - The number of primitive operations (steps) executed before termination
- Order of growth
  - The leading term of a formula
  - Expresses the behavior of a function toward infinity

Asymptotic Notations

- A way to describe behavior of functions in the limit
  - How we indicate running times of algorithms
  - Describe the running time of an algorithm as n grows to ∞
- O notation: asymptotic “less than”: f(n) “≤” g(n)
- Ω notation: asymptotic “greater than”: f(n) “≥” g(n)
- Θ notation: asymptotic “equality”: f(n) “=” g(n)

Logarithms

- In algorithm analysis we often use the notation "log n" without specifying the base

Binary logarithm $\lg n = \log_2 n$

Natural logarithm $\ln n = \log_e n$

$\lg^k n = (\lg n)^k$

$\lg \lg n = \lg(\lg n)$

$\log x^y = y \log x$

$\log xy = \log x + \log y$

$\log \frac{x}{y} = \log x - \log y$

$\log_a x = \log_a b \log_b x$

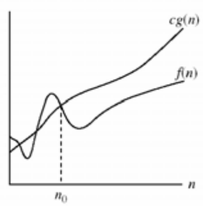
$a^{\log_b x} = x^{\log_b a}$

4

Asymptotic notations

- *O-notation*

$O(g(n)) = \{f(n) :$ there exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0\}$.



- Intuitively: *O*(g(n)) = the set of functions with a smaller or same order of growth as *g*(n)

$g(n)$ is an ***asymptotic upper bound*** for $f(n)$.

5

Examples

- $2n^2 = O(n^3)$: $2n^2 \le cn^3 \Rightarrow 2 \le cn \Rightarrow c = 1$ and $n_0 = 2$
- $n^2 = O(n^2)$: $n^2 \le cn^2 \Rightarrow c \ge 1 \Rightarrow c = 1$ and $n_0 = 1$
- $1000n^2+1000n = O(n^2)$:

$1000n^2+1000n \le 1000n^2+1000n^2=2000n^2 \Rightarrow c=2000$ and $n_0 = 1$

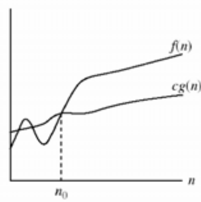
- $n = O(n^2)$: $n \le cn^2 \Rightarrow cn \ge 1 \Rightarrow c = 1$ and $n_0 = 1$

6

Asymptotic notations (cont.)

- Ω - *notation*

$\Omega(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$.



- Intuitively: $\Omega(g(n))$ = the set of functions with a larger or same order of growth as g(n)

$g(n)$ is an ***asymptotic lower bound*** for $f(n)$.

7

Examples

- $5n^2 = \Omega(n)$

  $\exists\ c, n_0$ such that: $0 \le cn \le 5n^2 \Rightarrow cn \le 5n^2 \Rightarrow c = 1$ and $n_0 = 1$

- $100n + 5 \ne \Omega(n^2)$

  $\exists\ c, n_0$ such that: $0 \le cn^2 \le 100n + 5$

  $100n + 5 \le 100n + 5n$ ($\forall\ n \ge 1$) = $105n$

  $cn^2 \le 105n \Rightarrow n(cn - 105) \le 0$

  Since n is positive $\Rightarrow cn - 105 \le 0 \Rightarrow n \le 105/c$

  $\Rightarrow$ contradiction: n cannot be smaller than a constant

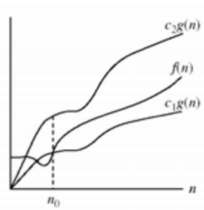
- $n = \Omega(2n)$, $n^3 = \Omega(n^2)$, $n = \Omega(\log n)$

8

Asymptotic notations (cont.)

- Θ-*notation*

$\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2, and n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0\}$.



- Intuitively $\Theta(g(n))$ = the set of functions with the same order of growth as g(n)

$g(n)$ is an ***asymptotically tight bound*** for $f(n)$.

9

Examples

- $n^2/2 - n/2 = \Theta(n^2)$
  - $\frac{1}{2} n^2 - \frac{1}{2} n \le \frac{1}{2} n^2 \ \forall n \ge 0 \quad \Rightarrow \quad c_2 = \frac{1}{2}$
  - $\frac{1}{2} n^2 - \frac{1}{2} n \ge \frac{1}{2} n^2 - \frac{1}{2} n * \frac{1}{2} n \ (\forall n \ge 2) = \frac{1}{4} n^2 \Rightarrow \quad c_1 = \frac{1}{4}$
- $n \ne \Theta(n^2)$: $c_1 n^2 \le n \le c_2 n^2 \Rightarrow$ only holds for: $n \le 1/c_1$
- $6n^3 \ne \Theta(n^2)$: $c_1 n^2 \le 6n^3 \le c_2 n^2 \Rightarrow$ only holds for: $n \le c_2 /6$
- $n \ne \Theta(\log n)$: $c_1 \log n \le n \le c_2 \log n$

  $\Rightarrow c_2 \ge n/\log n, \ \forall n \ge n_0$ – impossible

10

More on Asymptotic Notations

- There is no unique set of values for n_0 and c in proving the asymptotic bounds
- Prove that $100n + 5 = O(n^2)$
  - $100n + 5 \le 100n + n = 101n \le 101n^2$

    for all $n \ge 5$

    $n_0 = 5$ and $c = 101$ is a solution
  - $100n + 5 \le 100n + 5n = 105n \le 105n^2$

    for all $n \ge 1$

    $n_0 = 1$ and $c = 105$ is also a solution

Must find **SOME** constants c and n_0 that satisfy the asymptotic notation relation

11

Comparisons of Functions

- *Theorem:*

  $f(n) = \Theta(g(n)) \Leftrightarrow f = O(g(n))$ and $f = \Omega(g(n))$
- Transitivity:
  - $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$
  - Same for O and Ω
- Reflexivity:
  - $f(n) = \Theta(f(n))$
  - Same for O and Ω
- Symmetry:
  - $f(n) = \Theta(g(n))$ if and only if $g(n) = \Theta(f(n))$
- Transpose symmetry:
  - $f(n) = O(g(n))$ if and only if $g(n) = \Omega(f(n))$

12

Asymptotic Notations in Equations

- On the right-hand side
  - $\Theta(n^2)$ stands for some anonymous function in $\Theta(n^2)$

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

  There exists a function $f(n) \in \Theta(n)$ such that

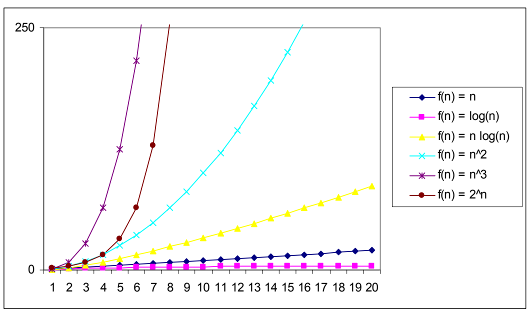
  $2n^2 + 3n + 1 = 2n^2 + f(n)$
- On the left-hand side

  $2n^2 + \Theta(n) = \Theta(n^2)$

  No matter how the anonymous function is chosen on the left-hand side, there is a way to choose the anonymous function on the right-hand side to make the equation valid.

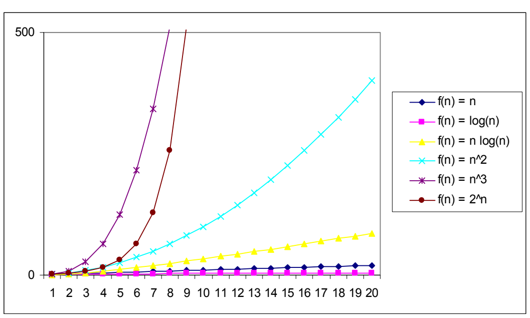
13

Practical Complexity



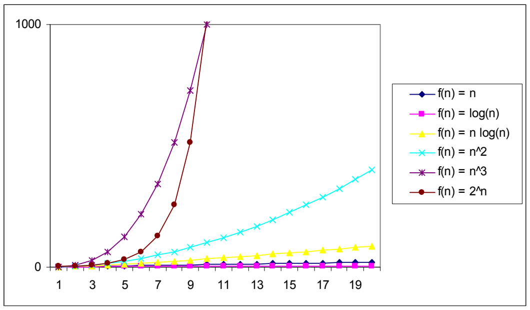
14

Practical Complexity

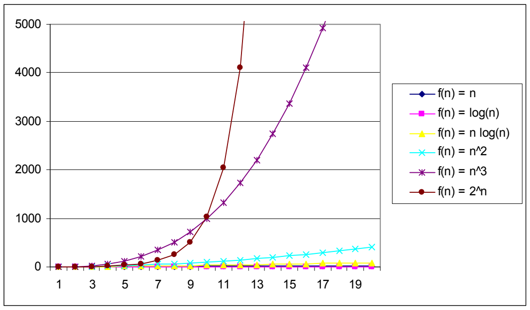


15

Practical Complexity



Practical Complexity



Asymptotic Notations - Examples

- For each of the following pairs of functions, either f(n) is O(g(n)), f(n) is Ω(g(n)), or f(n) = Θ(g(n)). Determine which relationship holds.
  - $f(n) = \log n^2$; $g(n) = \log n + 5$ — $f(n) = \Theta(g(n))$
  - $f(n) = n$; $g(n) = \log n^2$ — $f(n) = \Omega(g(n))$
  - $f(n) = \log \log n$; $g(n) = \log n$ — $f(n) = O(g(n))$
  - $f(n) = n$; $g(n) = \log^2 n$ — $f(n) = \Omega(g(n))$
  - $f(n) = n \log n + n$; $g(n) = \log n$ — $f(n) = \Omega(g(n))$
  - $f(n) = 10$; $g(n) = \log 10$ — $f(n) = \Theta(g(n))$
  - $f(n) = 2^n$; $g(n) = 10n^2$ — $f(n) = \Omega(g(n))$
  - $f(n) = 2^n$; $g(n) = 3^n$ — $f(n) = O(g(n))$

Other Asymptotic Notations

- A function f(n) is o(g(n)) if for any positive constant c, there exists n_0 such that
  $f(n) < c\, g(n) \ \forall\ n \geq n_0$

or $\lim_{n\to\infty} (f(n)/g(n)) = 0$

19

Other Asymptotic Notations

- A function f(n) is ω(g(n)) if for any positive constant c, there exists n_0 such that
  $c\, g(n) < f(n) \ \forall\ n \geq n_0$

or $\lim_{n\to\infty} (f(n)/g(n)) = \infty$

20

Intuitions

- Intuitively,
  - o() is like <
  - O() is like ≤
  - ω() is like >
  - Ω() is like ≥
  - Θ() is like =

21

Typical Running Time Functions

- Θ (1) (constant running time):
  - Instructions are executed once or a few times
- Θ (logN) (logarithmic)
  - A big problem is solved by cutting the original problem in smaller sizes, by a constant fraction at each step
- Θ (N) (linear)
  - A small amount of processing is done on each input element
- Θ (N logN)
  - A problem is solved by dividing it into smaller problems, solving them independently and combining the solution

22

Typical Running Time Functions

- Θ (N^2) (quadratic)
  - Typical for algorithms that process all pairs of data items (double nested loops)
- Θ (N^3) (cubic)
  - Processing of triples of data (triple nested loops)
- Θ (N^K) (polynomial)
- Θ (2^N) (exponential)
  - Few exponential algorithms are appropriate for practical use

23

Some Simple Summation Formulas

- Arithmetic series: $\sum_{k=1}^{n} k = 1 + 2 + ... + n = \frac{n(n+1)}{2}$
- Geometric series: $\sum_{k=0}^{n} x^k = 1 + x + x^2 + ... + x^n = \frac{x^{n+1} - 1}{x - 1} (x \neq 1)$
  - Special case: $x < 1$: $\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$
- Harmonic series: $\sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + ... + \frac{1}{n} \approx \ln n$
- Other important formulas: $\sum_{k=1}^{n} \lg k = \lg(k!) \approx n \lg n$

$$\sum_{k=1}^{n} k^p = 1^p + 2^p + ... + n^p \approx \frac{1}{p+1} n^{p+1}$$

24