Intro to Theory of Computation

Handout on

Asymptotic Notation

- O-, Ω-, Θ-, o-, ω-notation

Sofya Raskhodnikova

Asymptotic Order of Growth

- **Upper bound.** ***T(n)* is *O(f(n))*** if there exist positive integers c and n_0 such that for all $n \geq n_0$: $0 \leq T(n) \leq c \cdot f(n)$.
- **Lower bound.** ***T(n)* is Ω*(f(n))*** if there exist positive integers c and n_0 such that for all $n \geq n_0$: $T(n) \geq c \cdot f(n)$.
- **Tight bound.** ***T(n)* is Θ*(f(n))*** if T(n) is both O(f(n)) and Ω(f(n)).
- Example: $T(n) = 32n^2 + 17n + 32$.
  - T(n) is $O(n^2)$, $O(n^3)$, $\Omega(n^2)$, $\Omega(n)$, and $\Theta(n^2)$.
  - T(n) is not $O(n)$, $\Omega(n^3)$, $\Theta(n)$, or $\Theta(n^3)$.

Notation

- **One-sided equality:** T(n) = O(f(n)).
  - Not transitive:
    - $f(n) = 5n^3$; $g(n) = 3n^2$
    - $f(n) = O(n^3) = g(n)$
    - but $f(n) \neq g(n)$.
  - Alternative notation: $T(n) \in O(f(n))$.
- **Meaningless:** Any comparison-based sorting algorithm requires at least O(n log n) comparisons.
  - Use Ω for lower bounds.

Properties

- Transitivity.
  - If f = O(g) and g = O(h) then f = O(h).
  - If f = Ω(g) and g = Ω(h) then f = Ω(h).
  - If f = Θ(g) and g = Θ(h) then f = Θ(h).
- Additivity.
  - If f = O(h) and g = O(h) then f + g = O(h).
  - If f = Ω(h) and g = Ω(h) then f + g = Ω(h).
  - If f = Θ(h) and g = O(h) then f + g = Θ(h).

Common Functions: Asymptotic Bounds

- **Polynomials.** $a_0 + a_1 n + \ldots + a_d n^d$ is $\Theta(n^d)$ if $a_d > 0$.
- **Polynomial time.** Running time is $O(n^d)$ for some constant d independent of the input size n.
- **Logarithms.** $\log_a n = \Theta(\log_b n)$ for all constants a, b > 0.

  can avoid specifying the base

  log grows slower than every polynomial

  For every x > 0, $\log n = O(n^x)$.

  every exponential grows faster than every polynomial
- **Exponentials.** For all r > 1 and all d > 0, $n^d = O(r^n)$.
- **Factorial.** By Sterling's formula, $n! = 2^{\Theta(n \log n)}$

Overview

Notation	… means …	Think…	E.g.	Lim *f(n)/g(n)*
$f(n)=O(n)$	$\exists\, c>0, n_0>0, \forall\, n > n_0$: $0 \leq f(n) < cg(n)$	Upper bound	$100n^2 = O(n^3)$	If it exists, it is $< \infty$
$f(n)=\Omega(g(n))$	$\exists\, c>0, n_0>0, \forall\, n > n_0$: $0 \leq cg(n) < f(n)$	Lower bound	$n^{100} = \Omega(2^n)$	If it exists, it is > 0
$f(n)=\Theta(g(n))$	both of the above: $f=\Omega(g)$ **and** $f = O(g)$	Tight bound	$\log(n!) = \Theta(n \log n)$	If it exists, it is > 0 and < ∞
$f(n)=o(g(n))$	$\forall\, c>0, n_0>0, \forall\, n > n_0$: $0 \leq f(n) < cg(n)$	Strict upper bound	$n^2 = o(2^n)$	Limit exists, =0
$f(n)=\omega(g(n))$	$\forall\, c>0, n_0>0, \forall\, n > n_0$: $0 \leq cg(n) < f(n)$	Strict lower bound	$n^2 = \omega(\log n)$	Limit exists, $=\infty$

Some functions sorted by asymptotic growth

- $\log n$
- $\sqrt{n}$
- n
- $n \log n$
- n^2
- $n^{1,000,000}$
- 2^n (beats n^k for any fixed k)
- $n!$

S. Raskhodnikova; based on slides by K. Wayne, E. Demaine, C. Leiserson, A. Smith.