Asymptotic Notations

Fahad Panolan

Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India

6 January 2022

Recall

- Input length
- RAM model - Each instruction takes unit time. Read/Write from a memory location take unit cost.
- Worst-case running time, best-case running time.
- Correctness proof: Loop invariant and mathematical induction.

Run Time Analysis (Time Complexity)

- The number of instructions executed by the algorithms. This may be different for different inputs.
- Each execution of each instruction takes **one unit** of time. (We assume RAM model)

Worst-case and best-case running time

- Let $\mathcal{A}$ be an algorithm for a problem Π and $R(x)$ be its running time on input x.
- Worst-case running time of is the function $T \colon \mathbb{N} \mapsto \mathbb{N}$:

$$T(n) = \max_{x \colon |x|=n} R(x).$$

Worst-case and best-case running time

- Let $\mathcal{A}$ be an algorithm for a problem Π and $R(x)$ be its running time on input x.

- Worst-case running time of is the function $T\colon \mathbb{N} \mapsto \mathbb{N}$:

$$T(n) = \max_{x\colon |x|=n} R(x).$$

That is, for any input y of "length n", $\mathcal{A}$ on y executes at most $T(n)$ instructions.

Worst-case and best-case running time

- Let $\mathcal{A}$ be an algorithm for a problem Π and $R(x)$ be its running time on input x.

- Worst-case running time of is the function $T \colon \mathbb{N} \mapsto \mathbb{N}$:

$$T(n) = \max_{x \colon |x|=n} R(x).$$

  That is, for any input y of "length n", $\mathcal{A}$ on y executes at most $T(n)$ instructions.

- Best-case running time of $\mathcal{A}$ is the function $T_b \colon \mathbb{N} \mapsto \mathbb{N}$:

$$T_b(n) = \min_{x \colon |x|=n} R(x).$$

For Insertion-Sort

- $T(n) = \frac{3}{2}n^2 - \frac{5}{2}n + 3$ $\qquad T_b(n) = 5n - 5$

Comparing different algorithms

Processor performance: 1 million instructions per second.

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long

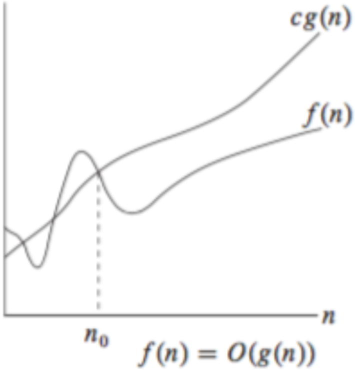
Figure from "Algorithm Design" by Kleinberg and Tardos

Asymptotic Notations

big-O

$O(g(n)) = \{f(n) : \exists\ c, n_0 > 0 \text{ s.t. } \forall n \geq n_0,\ 0 < f(n) \leq cg(n)\}$

- $O(\cdot)$ is used to asymptotically upper bound a function.
- We think of $f(n) \in O(g(n))$ as "$f(n) \leq g(n)$".



All the functions considered here (in the definitions of asymptotic notations) are asymptotically positive.

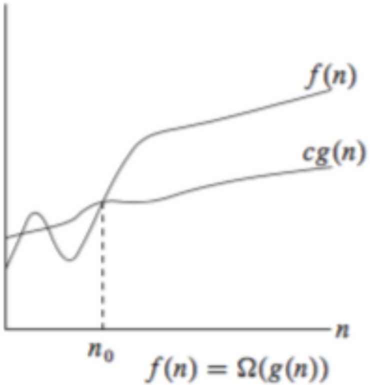
Examples

$f_1(n) = 10n^2$, $f_2(n) = n^2$, $f_3(n) = 1000n$

big-Omega(Ω)

$\Omega(g(n)) = \{f(n) : \exists\ c, n_0 > 0$ s.t. $\forall n \geq n_0,\ f(n) \geq cg(n) > 0\}$

- $\Omega(\cdot)$ is used to asymptotically lower bound a function.
- We think of $f(n) \in \Omega(g(n))$ as "$f(n) \geq g(n)$".



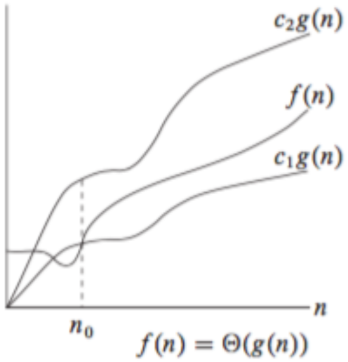
Examples

$f_1(n) = 10n^2$, $f_2(n) = n^2$, $f_3(n) = 1000n$

Theta(Θ)

$\Theta(g(n)) =$
$\{f(n) : \exists\ c_1, c_2, n_0 > 0$ s.t. $\forall n \geq n_0,\ c_1 g(n) \leq f(n) \leq c_2 g(n)\}$

- $f(n) \in \Theta(n)$ iff $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.
- We think of $f(n) \in \Omega(g(n))$ as "$f(n) = g(n)$".



Examples

$f_1(n) = 10n \log n + \sqrt{n} \log^2(n)$, $f_2(n) = n \log n$.

Asymptotic analysis: INSERTION SORT

```
INSERTION-SORT(A)
for j = 2 to A.length
    key = A[j]
    // Insert A[j] into the sorted sequence A[1..j - 1].
    i = j - 1
    while i > 0 and A[i] > key
        A[i + 1] = A[i]
        i = i - 1
    A[i + 1] = key
```

- Worst-case running time, $T(n) = \frac{3}{2}n^2 - \frac{5}{2}n + 3$
- Best-case running time $T_b(n) = 5n - 5$

More Asymptotic Notations

Little-o

$o(g(n)) =$
$\{f(n) : \forall\ c > 0 \ \ \exists\ n_0 > 0$ s.t. $\ \forall n \geq n_0,\ 0 < f(n) < cg(n)\}$

- We think of $f(n) \in o(g(n))$ as "$f(n) < g(n)$".

Little-omega

$\omega(g(n)) =$
$\{f(n) : \forall\ c > 0\ \ \exists\ n_0 > 0$ s.t. $\ \forall n \geq n_0,\ f(n) > cg(n) > 0\}$

- We think of $f(n) \in o(g(n))$ as "$f(n) > g(n)$".

Examples

Order the following functions from fastest to slowest:

$$\sqrt{2}^{\lg n}, n^2, (\frac{3}{2})^n, \lg n^2, \lg^2 n, 2^n, \lg\lg n, n \lg n$$

Thank You.