CSC006P1M: Design and Analysis of Algorithms
Lecture 04 *(Analysis of Algorithms)*

Sumit Kumar Pandey

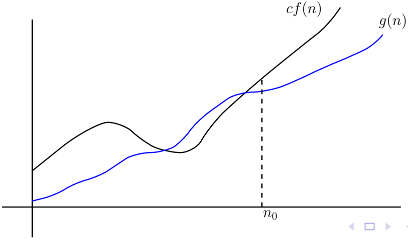
August 22, 2022

Asymptotic Notations

The O Notation

We say that a function $g(n)$ is $O(f(n))$ (pronounced "big oh" of $f(n)$), if **there exist** positive constants c and n_0, such that for all $n \geq n_0$, we have $0 \leq g(n) \leq cf(n)$.

- In other words, $f(n)$ is an asymptotically upper bound for $g(n)$. We write $g(n) = O(f(n))$ or $g(n) \in O(f(n))$.
- Informally, $f(n)$ describes the upper bound for $g(n)$.
- The O notation bounds $g(n)$ only from above.



Some Examples

- $100n = O(n)$
- $100n = O(5n + 6)$
- $100n = O(23n)$
- $100n = O(n^2)$
- $100n = O(n^3)$
- $2n^2 + 50 = O(n^2)$
- $2n^2 + 50 = O(7n^2 + 31)$
- $2n^2 + 50 = O(n^2 + 7132)$
- $2n^2 + 50 = O(n^3)$
- $2n^2 + 50 = O(n^4)$
- $2n^2 + 50 \neq O(n)$
- $6n^3 + 4n^2 + 5 \neq O(n^2)$
- $constant = O(1)$
- $constant = O(n)$

Upper, Lower and Exact Bounds

- Upper bound is a crude bound.
- We are interested in finding the expression which is as close to the actual running time as possible.
- In cases, where finding an exact expression is difficult, we are interested in finding upper bound of the running time.

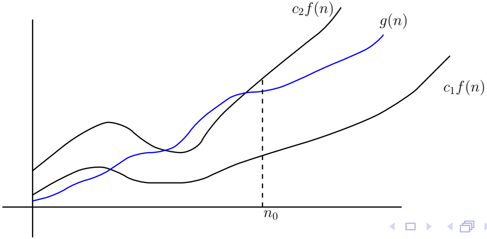
But, why not lower bound of the running time of an algorithm?

In-fact, we are more interested in finding the lower bound of the **problem**, rather than lower bound of the running time of any **algorithm** that solves the problem. But, finding the lower bound of the problem is not easy.

The Θ Notation

We say that a function $g(n)$ is $\Theta(f(n))$ (pronounced "theta" of $f(n)$), if **there exist** positive constants c_1, c_2 and n_0, such that for all $n \geq n_0$, we have $0 \leq c_1 f(n) \leq g(n) \leq c_2 f(n)$.

- In other words, $f(n)$ is an asymptotically tight bound for $g(n)$. We write $g(n) = \Theta(f(n))$ or $g(n) \in \Theta(f(n))$.
- Informally, $f(n)$ describes the exact bound for $g(n)$.
- The Θ notation bounds $g(n)$ from both above and below.



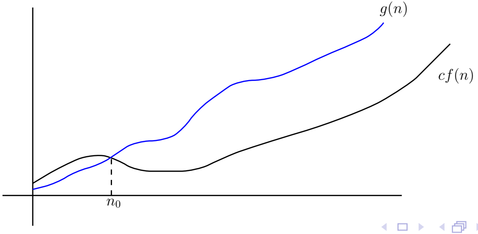
Some Examples

- $100n = \Theta(n)$
- $100n = \Theta(5n+6)$
- $100n = \Theta(23n)$
- $100n \neq \Theta(n^2)$
- $100n \neq \Theta(n^3)$
- $2n^2 + 50 = \Theta(n^2)$
- $2n^2 + 50 = \Theta(7n^2 + 31)$
- $2n^2 + 50 = \Theta(n^2 + 7132)$
- $2n^2 + 50 \neq \Theta(n^3)$
- $2n^2 + 50 \neq \Theta(n^4)$
- $2n^2 + 50 \neq \Theta(n)$
- $6n^3 + 4n^2 + 5 \neq \Theta(n^2)$
- $constant = \Theta(1)$
- $constant \neq \Theta(n)$

The Ω *Notation*

We say that a function $g(n)$ is $\Omega(f(n))$ (pronounced "big omega" of $f(n)$), if **there exist** positive constants c and n_0, such that for all $n \geq n_0$, we have $0 \leq cf(n) \leq g(n)$.

- In other words, $f(n)$ is an asymptotically lower bound for $g(n)$. We write $g(n) = \Omega(f(n))$ or $g(n) \in \Omega(f(n))$.
- Informally, $f(n)$ describes the lower bound for $g(n)$.
- The Ω notation bounds $g(n)$ only from below.



Some Examples

- $100n = \Omega(n)$
- $100n = \Omega(5n + 6)$
- $100n = \Omega(23n)$
- $100n \neq \Omega(n^2)$
- $100n \neq \Omega(n^3)$
- $2n^2 + 50 = \Omega(n^2)$
- $2n^2 + 50 = \Omega(7n^2 + 31)$
- $2n^2 + 50 = \Omega(n^2 + 7132)$
- $2n^2 + 50 \neq \Omega(n^3)$
- $2n^2 + 50 \neq \Omega(n^4)$
- $2n^2 + 50 = \Omega(n)$
- $6n^3 + 4n^2 + 5 = \Omega(n^2)$
- $constant = \Omega(1)$
- $constant \neq \Omega(n)$

The o Notation

We say that a function $g(n)$ is $o(f(n))$ (pronounced "little oh" of $f(n)$), if **for any** positive constant c, **there exists** a positive constant n_0, such that for all $n \geq n_0$, we have $0 \leq g(n) < cf(n)$.

Alternatively,

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

- We write $g(n) = o(f(n))$ or $g(n) \in o(f(n))$.
- The asymptotic upper bound provided by O notation **may or may not** be asymptotically tight.
- The o notation is used to denote an upper bound that is **not** asymptotically tight.

Some Examples

- $100n \neq o(n)$
- $100n \neq o(5n + 6)$
- $100n \neq o(23n)$
- $100n = o(n^2)$
- $100n = o(n^3)$
- $2n^2 + 50 \neq o(n^2)$
- $2n^2 + 50 \neq o(7n^2 + 31)$
- $2n^2 + 50 \neq o(n^2 + 7132)$
- $2n^2 + 50 = o(n^3)$
- $2n^2 + 50 = o(n^4)$
- $2n^2 + 50 \neq o(n)$
- $6n^3 + 4n^2 + 5 \neq o(n^2)$
- $constant \neq o(1)$
- $constant = o(n)$

The ω Notation

We say that a function $g(n)$ is $\omega(f(n))$ (pronounced "little omega" of $f(n)$), if **for any** positive constant c, **there exists** a positive constant n_0, such that for all $n \geq n_0$, we have $0 \leq cf(n) < g(n)$.

Alternatively,

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

- We write $g(n) = \omega(f(n))$ or $g(n) \in \omega(f(n))$.
- The asymptotic lower bound provided by Ω notation **may or may not** be asymptotically tight.
- The ω notation is used to denote a lower bound that is **not** asymptotically tight.

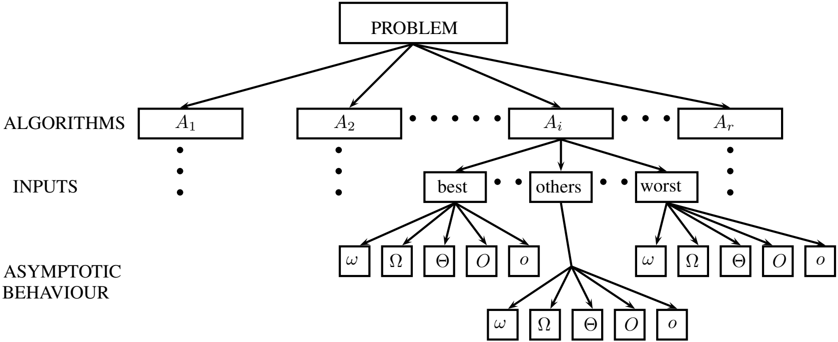
Some Examples

- $100n \neq \omega(n)$
- $100n \neq \omega(5n + 6)$
- $100n \neq \omega(23n)$
- $100n \neq \omega(n^2)$
- $100n \neq \omega(n^3)$
- $2n^2 + 50 \neq \omega(n^2)$
- $2n^2 + 50 \neq \omega(7n^2 + 31)$
- $2n^2 + 50 \neq \omega(n^2 + 7132)$
- $2n^2 + 50 \neq \omega(n^3)$
- $2n^2 + 50 \neq \omega(n^4)$
- $2n^2 + 50 = \omega(n)$
- $6n^3 + 4n^2 + 5 = \omega(n^2)$
- $constant \neq \omega(1)$
- $constant \neq \omega(n)$

How to Remember?

$g(n) = \omega(f(n))$	$f(n) < g(n)$
$g(n) = \Omega(f(n))$	$f(n) \leq g(n)$
$g(n) = \Theta(f(n))$	$f(n) = g(n)$
$g(n) = O(f(n))$	$g(n) \leq f(n)$
$g(n) = o(f(n))$	$g(n) < f(n)$

Summary



PROBLEM
ALGORITHMS
A_1
A_2
A_i
A_r
INPUTS
best
others
worst
ASYMPTOTIC BEHAVIOUR
ω Ω Θ O o
ω Ω Θ O o
ω Ω Θ O o

Thank You