Lecture 4: Asymptotic Notation Introduction

CS303: Algorithms

January 13, 2014

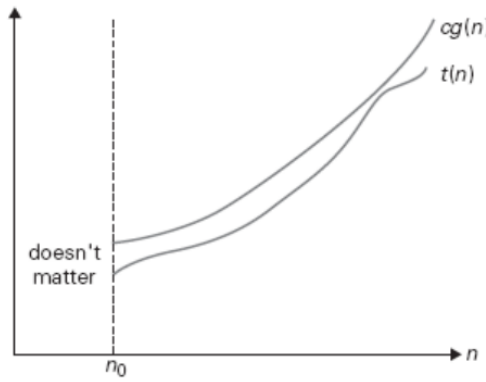
1 Asymptotic Notations

1.1 O notation

O(g(n)): a set of functions with the same or smaller order of growth as $g(n) \rightarrow t(n) \leq g(n)$

- $2n^2 - 5n + 1 \in O(n^2)$
- $2n + n^{100} - 2 \in O(n!)$
- $2n + 6 \notin O(logn)$

Formal: There exists a non-negative constant c and a non-negative integer n_0 such that $t(n) \leq cg(n)$ for all $n \geq n_0$
i.e. when n is large enough, g(n) is always bigger than f(n) $\rightarrow$ big-O is the asymptotic upper bound



Big-oh notation: $t(n) \in O(g(n))$.

Example: Prove $10n \in O(n^2)$

Let $n_0 = 10, c = 1$, then $10n \leq n^2$ for all $n \geq n_0$

Sometimes, we write that $f(n) = O(g(n))$ where = really means "is". Thus O notation is not symmetrical. e.g. $n^2 = O(n^3)$, not the other way around.
Q:/ What is the use of Big O notation?
A:/ Most of time it is used as error bound.

e.g. $T(n) = n^3 + O(n^2)$. This means that there is a function $h(n) \in O(n^2)$ such that $T(n) = n^3 + h(n)$. It is better than just say $T(n) = n^3$

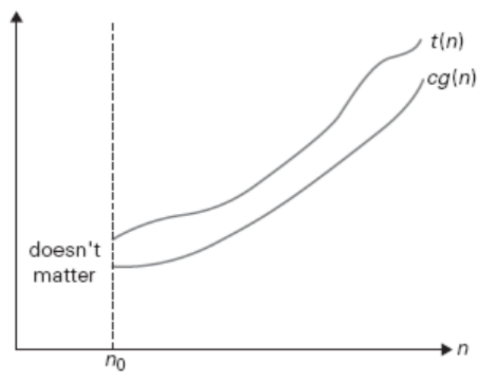
e.g. $n^2 + O(n) = O(n^2)$. Means for any $f(n)$ in $O(n)$, there is an $h(n)$ in $O(n^2)$ such that $n^2 + f(n) = h(n)$
Q:/ Can we say the other way around?
A:/ no

1.2 Ω notation

$\Omega(g(n))$: a set of functions with the same or larger order of growth as $g(n) \rightarrow t(n) \geq g(n)$

- $2n^2 - 5n + 1 \in \Omega(n)$
- $2n + n^{100} - 2 \notin \Omega(n!)$
- $2n + 6 \in \Omega(logn)$

Formal: There exists a non-negative constant c and a non-negative integer n_0 such that $t(n) \geq cg(n)$ for all $n \geq n_0$



Big-omega notation: $t(n) \in \Omega(g(n))$.

Example
$n^2 \in \Omega(logn)$
let $n_0 = 1, c = 1$, then $n^2 \geq log(n)$ for all $n \geq n_0$

Example
$\sqrt{n} \in \Omega(lgn)$

1.3 Θ notation

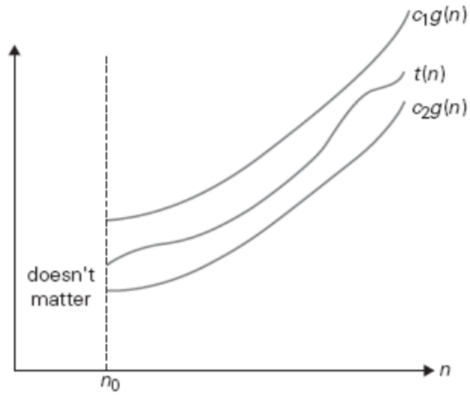
$\Theta(g(n))$: a set of functions with the same or larger order of growth as $g(n) \rightarrow t(n) == g(n)$

- $2n^2 - 5n + 1 \in \Theta(n^2)$

- $2n + n^{100} - 2 \notin \Theta(n!)$
- $2n + 6 \notin \Theta(logn)$

Formal: there exist positive constant c_1 and c_2 and non-negative integer n_0 such that $c_1 g(n) \leq t(n) \leq c_2 g(n)$ for every $n \geq n_0$.

e.g. $2n - 51 \in \Theta(n)$ (for$c_1 = 1, c_2 = 2, n_0 = 7$)



Big-theta notation: $t(n) \in \Theta(g(n))$.

In a sense, $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

Note:

- There are also small o and small ω notation that represents $<$ and $>$ respectively. Be careful here in small o and small ω, the relation must be true for any constant c

  e.g. To prove $2n^2 = o(n^3)$ we can find n_0 as $\frac{2}{c}$ such that for any c, $2n^2$ is always less than n^3

  e.g.$\frac{1}{2}n^2$ is in $O(n^2)$, but not in $o(n^2)$
- Using asymptotic notations doesn't mean that the constant factors are not important. Also asymptotic notations can be used for all three cases!
- Some of the common major terms in T(n) from "smallest" to "largest"
  $c, log(n), \sqrt{n}, n, nlog(n), n^2, n^3, 2^n, 3^n, n!$

Exercise: Determine whether the following assertions are true or false

$$
\begin{aligned}
\frac{n(n+1)}{2} &\in O(n^3) \\
\frac{n(n+1)}{2} &\in O(n^2) \\
\frac{n(n+1)}{2} &\in \Theta(n^3) \\
\frac{n(n+1)}{2} &\in \Omega(n^2) \\
n^2 + n + log(n) + 1 &\in \Theta(n^2) \\
n^2 + n + log(n) + 1 &\in \Theta(n^3)
\end{aligned}
$$

The answers are T, T, F, T, T, F