JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Chapter 3
Growth of Functions

Outline

- Asymptotic Notation
  - O-notation
  - Ω-notation
  - Θ-notation

Overview

- A way to describe behavior of function in the limit.
- Describing growth of functions.
- Focus on what is important by abstracting away low-order terms and constant factors.
- Ow running times of algorithms is indicated.

Asymptotic Notation

- O-notation (upper bounds):

- We write $f(n) = O(g(n))$ if there exist constants $c > 0$, $n_0 > 0$ such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$.

Example: $2n^2 = O(n^3)$ $\quad (c = 1, n_0 = 2)$

functions, not values

This is a one-way equality

Set Definition of O-notation

- $O(g(n)) = \{ f(n)$: there exist constants $c > 0$, $n_0 > 0$ such that $0 \leq f(n) \leq cg(\mathrm{n})$ for all $n \geq n_0\}$

Example: $2n^2 \in O(n^3)$

Note: if $n2n^2 \in O(n^3)$, may be an unconventional representation, however it is important to know how the notation is to be represented.

Asymptotic Notation in Equations

- When on the right-hand side

$O(n^2)$ stands for some anonymous function in the set $O(n^2)$.

$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ means $2n^2 + 3n + 1 = 2n^2 + f(n)$ for some $f(n) \in \Theta(n)$. In particular, $f(n) = 3n + 1$.

By the way, we interpret # of anonymous functions as = # of times the asymptotic notation appears:

$\sum_{i=1}^{n} O(i)$ OK: 1 anonymous functions

$O(1) + O(2) + \ldots + O(n)$ not OK: n hidden constants

=> no clean interpretation

Asymptotic Notation in Equations cont.

- When on the left-hand side

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

Interpret $2n^2 + \Theta(n) = \Theta(n^2)$ as meaning for all functions $f(n) \in \Theta(n)$, there exists a function $g(n) \in \Theta(n^2)$ such that $2n^2 + f(n) = g(n)$.

Can chain together:

$$\begin{aligned} 2n^2 + 3n + 1 &= 2n^2 + \Theta(n) \\ &= \Theta(n^2) \end{aligned}$$

Asymptotic Notation in Equations cont.

Interpretation

- First equation: There exists $f(n) \in \Theta(n)$ such that $2n^2 + 3n + 1 = 2n^2 + f(n)$.
- Second equation: For all $g(n) \in \Theta(n)$ (such as the $f(n)$ used to make the first equation hold), there exists $h(n) \in \Theta(n^2)$ such that $2n^2 + g(n) = h(n)$.

Ω-notation (Lower Bounds)

- O-notation is an *upper-bound* notation. It makes no sense to say $f(n)$ is at least $O(n^2)$.
- $\Omega(g(n)) = \{ f(n)$: there exist constants $c > 0$, $n_0 > 0$ such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0\}$

Example: $\sqrt{n} = \Omega(\lg n)$ $\quad (c = 1, n_0 = 16)$

Θ-notation (Tight Bounds)

- $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

Example: $n^2 - 2n = \Theta\,(n^2)$

o-notation and ω-notation

- O-notation and Ω-notation are like ≤and ≥.
- o-notation and ω-notation are like < and >.
- $o(g(n))$= { $f(n)$: for any constant $c > 0$, there is a constant $n_0 > 0$ such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0$ }

  Example: $2n^2 = o(n^3)$ $(n_0 = 2/c)$

- $\omega(g(n))$ = { $f(n)$: for any constant $c > 0$, there is a constant $n_0 > 0$ such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0$}

  Example: $\sqrt{n} = \omega(\lg n)$ $(n_0 = 1 + 1/c)$

References

- T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "*Introduction to Algorithms*, *Third Edition*", 2009
- MIT Prof. Erik D. Demaine and MIT Prof. Charles E. Leiserson, Lecture Notes, Slides and Videos