Asymptotic Notations

Introduction

- The order of growth of the running time of an algorithm gives a simple characterization of the algorithm’s efficiency.

- Also allows us to compare the relative performance of alternative algorithms.

- Mostly, we are concerned with how the running time of an algorithm increases with the size of the input in the limit.

- Usually, an algorithm that is asymptotically more efficient will be the best choice for all but very small inputs.

Asymptotic Notation

- Asymptotic notation describes the running times of algorithms.

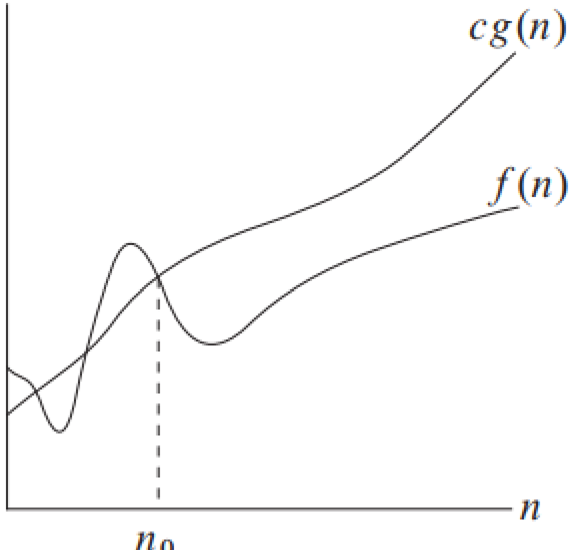
- When we use asymptotic notation to apply to the running time of an algorithm, we need to understand *which* running time we mean.
  - Worst-case running time
  - Average-case running time
  - Best-case running time

- Often one wish to make a blanket statement that covers all inputs, not just the worst case.

Big-Oh Notation

- When we have only an **asymptotic upper bound**, we use O-notation.

- For a given function g(n), we denote by O(g(n)) (pronounced "big-oh of g of n" or sometimes just "oh of g of n") the set of functions



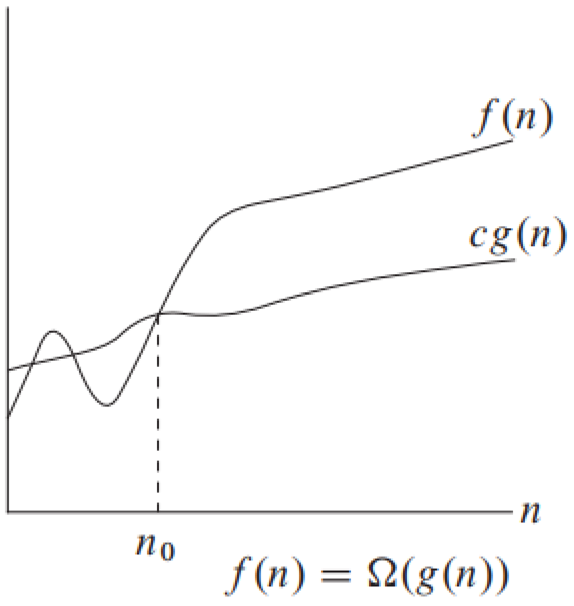
$f(n) = O(g(n))$

$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0\}.$$

Big-Omega Notation

- Just as O-notation provides an asymptotic upper bound on a function, Ω-notation provides an **asymptotic lower bound**.

- For a given function g(n), we denote by Ω(g(n)) (pronounced "big-omega of g of n" or sometimes just "omega of g of n") the set of functions



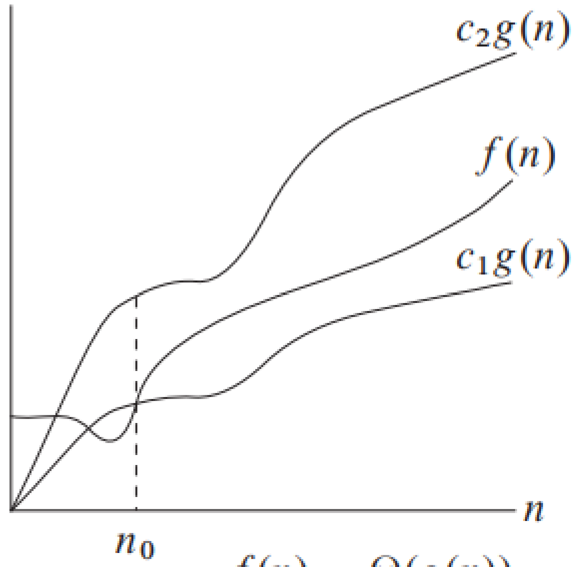
$$\Omega(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0 \} .$$

Theta Notation

- For a given function g(n), we denote by θ(g(n)) the set of functions:

$$\Theta(g(n)) = \{ f(n) : \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \} .$$[1]

- A function f(n) belongs to the set, (g(n)) if there exist positive constants c_1 and c_2 such that it can be “sandwiched” between $c_1g(n)$ and $c_2g(n)$, for sufficiently large n.



$f(n) = \Theta(g(n))$

Example: Big-Oh

Show that $n^2/2 - 3n = O(n^2)$

- First determine positive constants c_1, and n_0 such that

$$n^2/2 - 3n \le c_1 n^2 \text{ for all } n \ge n_0$$

- Diving by n^2, we have: $1/2 - 3/n \le c_1$
- For: **n=0**, $½ - 3/0 \le c_1$ (Not Holds)

  n=1, $½ - 3/1 \le c_1$ (Holds for $c_1 \ge ½$)

  n=2, $½ - 3/2 \le c_1$ (Holds and so on…)
- Considering Right Hand Side Inequality, it holds for any value $n \ge 1$ and constant $c_1 \ge ½$.
- Thus by choosing the constant $\mathbf{c_1 \ge ½}$ and $\mathbf{n_0 \ge 1}$, one can verify that $n^2/2 - 3n = O(n^2)$ holds.

Example: Big-Omega

Show that $n^2/2 - 3n = \Omega(n^2)$

- First determine positive constants c_1, and n_0 such that

$$c_1 n^2 <= n^2/2 - 3n \quad \text{for all } n >= n_0$$

- Diving by n^2, we have: $c_1 <= 1/2 - 3/n$
- For: **n=0**, $c_1 <= ½ - 3/0$ (Not Holds) **n=1**, $c_1 <= ½ - 3/1$ (Not Holds)

  n=2, $c_1 <= ½ - 3/2$ (Not Holds) **n=3**, $c_1 <= ½ - 3/3$ (Not Holds)

  n=4, $c_1 <= ½ - 3/4$ (Not Holds) **n=5**, $c_1 <= ½ - 3/5$ (Not Holds)

  n=6, $c_1 <= ½ - 3/6$ (Not Holds and Equals ZERO)

  n=7, $c_1 <= ½ - 3/7$ *or* $c_1 <= (7-6)/14$ *or* $c_1 <= 1/14$ (Holds for $c_1 <= 1/14$)
- Considering Left Hand Side Inequality, it holds for any value $n >= 7$ and constant $c_1 <= 1/14$.
- Thus by choosing the constant $\mathbf{c_1 <= 1/14}$ and $\mathbf{n_0>=7}$, one can verify that $n^2/2 - 3n = \Omega(n^2)$ holds.

Example: Theta

Show that $n^2/2 - 3n = \theta(n^2)$

- First determine positive constants c_1, c_2, and n_0 such that

$$c_1 n^2 <= n^2/2 - 3n <= c_2 n^2 \text{ for all } n >= n_0$$

- Diving by n^2, we have:

$$c_1 <= 1/2 - 3/n <= c_2$$

- Considering Right Hand Side Inequality, it holds for any value $n >= 1$ and constant $c_2 >= ½$.
- Considering Left Hand Side Inequality, it holds for any value $n >= 7$ and constant $c_1 <= 1/14$.
- Thus by choosing the constants $\mathbf{c_1 <= 1/14}$ and $\mathbf{c_2}$ **>= ½** and $\mathbf{n_0 >= 7}$, one can verify that $n^2/2 - 3n = \theta(n^2)$ holds.

o-Notation

- o-notation denotes an upper bound that is not asymptotically tight.
- Formally o(g(n)) ("little-oh of g of n") is defined as the set

$$o(g(n)) = \{ f(n) : \text{ for any positive constant } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le f(n) < cg(n) \text{ for all } n \ge n_0 \} .$$

- For example, $2n = o(n^2)$, but $2n^2 \mathrel{!}= o(n^2)$.
- Intuitively, in o-notation, the function f(n) becomes insignificant relative to g(n) as n approaches infinity; that is, $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

ω-Notation

- ω-notation denotes a lower bound that is not asymptotically tight.
- One way to define it is by:

$$f(n) \in \omega(g(n)) \text{ if and only if } g(n) \in o(f(n))$$

- Formally, ω(g(n)) ("little-omega of g of n") is defined as the set

$$\omega(g(n)) = \{ f(n) : \text{for any positive constant } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le cg(n) < f(n) \text{ for all } n \ge n_0 \} .$$

- For example, $n^2/2 = \omega(n)$, but $n^2/2 \;!= \omega(n^2)$.
- The relation $f(n) \in \omega(g(n))$ implies that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, if limit exists.

Reference

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.

Thank You