Data Structures and Algorithm Analysis

12

Dr. Syed Asim Jalal
Department of Computer Science
University of Peshawar

In this lecture ….

- Asymptotic Performance
  - How does the algorithm behave as the problem size gets very large…. ?

- Asymptotic Notations
  - O
  - Θ
  - Ω
  - o
  - ω

What is asymptotic analysis ?

- Asymptotic analysis deals with analyzing the properties of the running time when the input size goes to infinity (this means a very large input size)
    - The differences between orders of growths are more significant for larger input size. Analyzing the running times on small inputs does not allow us to distinguish between efficient and inefficient algorithms
  - The objective of asymptotic analysis is to describe the behavior of a function *T(N)* as it goes to infinity.
  - Asymptotic notations are used to describe the asymptotic analysis

Function Bounds..

- Lets understand with the help of example. Suppose we have a function $10N^2$
- Can we say it is bounded by $11N^2$ and $9N^2$ for all $N \geq 1$?
  - i.e $10N^2$ cannot go above $11N^2$ and doesn't come down below $9N^2$ for all values of N. $10N^2$ is sandwiched between $9N^2$ and $11N^2$
  - Now if f(n) is $10N^2$ and g(n) is N^2
  - Then we say that **f(n) is Θ (g(n))**
    - (explanation later..)

Asymptotic Notations

big-Theta $\Theta(g(n))$

Big-Oh $O(g(n))$

big-Omega $\Omega(g(n))$

little-oh $o(g(n))$

little-omega $\omega(g(n))$

Big-Theta Θ

- A function f(n) is Θ (g(n)) if there exist a positive constants c_1, c_2, and n_0 such that

$$0 \le c_1\, g(n) \le f(n) \le c_2\, g(n), \textit{ for all } n \ge n_0$$

  – We define Θ(g(n)) to be a ***set*** of functions that are ***asymptotically equivalent*** to g(n)

  – A function f(n) belongs to the set Θ(*g(n)*), if there exist positive constants c_1 and c_2, such that g(n) can be "sandwiched" between c_1g(n) and c_2g(n), for sufficiently large n.

- Representation:
  - "$f(n) = \Theta(g(n))$" or
  - "$f(n) \in \Theta(g(n))$"

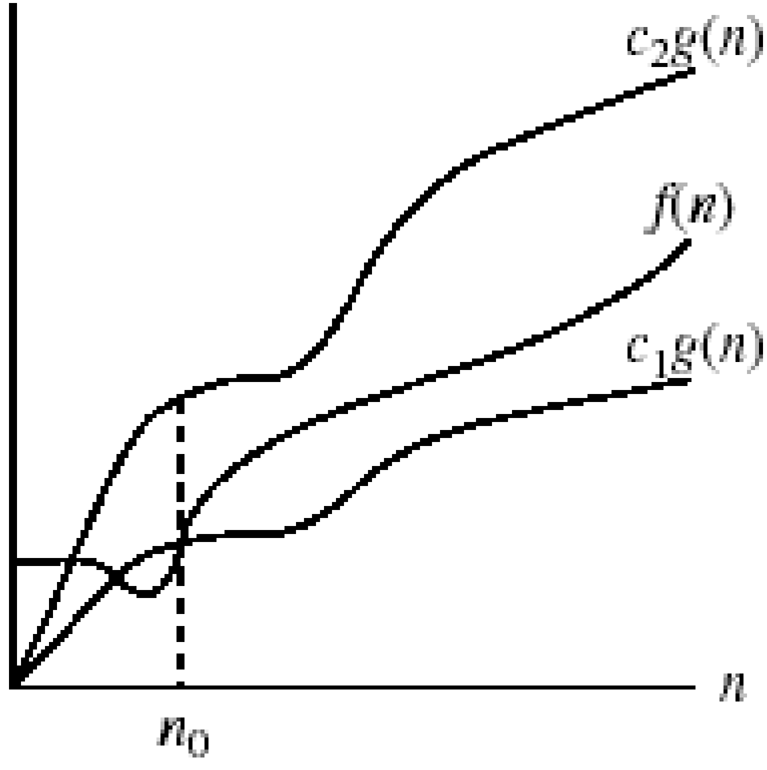
- We say this as:
  - f(n) and g(n) are ***asymptotically equivalent***.

  Or

  - g(n) is ***asymptotically tight bound*** *for f(n)*

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

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$



- The following equations are asymptotically equivalent
- $5n^2$
- $2n^2 - 5n + 10$
- $(8n^2 + 2n – 3)$
- $(n^2/5 + \sqrt{n} – 10 \log n)$
- $n(n – 3)$

As ‘n’ becomes large, the **dominant** term is some constant times $\mathbf{n^2}$

Lower and upper bounds *(Example1)*

- $f(n) = 8n^2 + 2n - 3$
  - To show that $f(n) \in \Theta(n^2)$
  - We need to find the following three values.
  - $c1$, $c2$ and n_o
- **To find Lower bound we need c1 and n_o**
- **To find Upper bound we need c2 and n_o**
  - **We will have two n_o, select the maximum n_o**

Finding c_1 and n_o *(Example1)*

Lower bound: $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$

$f(n) = 8n^2 + 2n - 3$, $f(n) \in \Theta(n^2)$

- $c_1 n^2 \le 8n^2 + 2n - 3$??
  - $7n^2 \le 8n^2 + 2n - 3$
  - $c_1 = 7$
  - $N_o = 1$

c_1 can be anything lesser than the constant with n^2 of the expression

$n_o = 1$

$7(1)^2 \le 8(1)^2 + 2(1) - 3$

$7 \le 8+2-3$

$7 \le 7$

Finding c_2 and n_o *(Example1)*

Upper Bound: $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$

$f(n) = 8n^2 + 2n - 3$, $f(n) \in \Theta(n^2)$

$8n^2 + 2n - 3 \leq c_2 n^2$

$8n^2 + 2n - 3 \leq 9n^2$

$\Rightarrow 8n^2 + 2n - 3 \leq 9n^2$

c_2 can be anything greater than the constant with n^2 of the expression

$c_2 = 9$

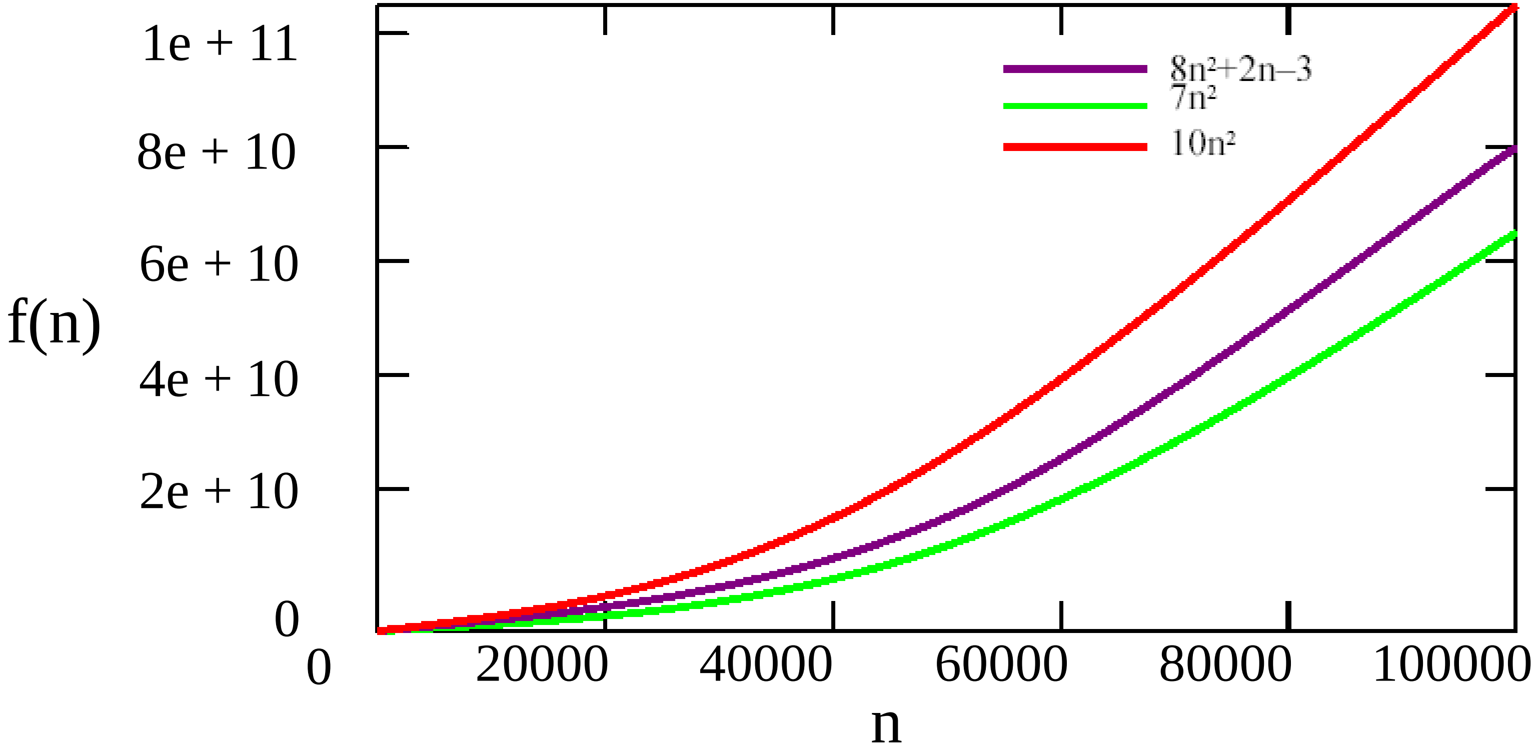
$N_o = 1$

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

$f(n) = 8n^2 + 2n - 3$

$g(n) = n^2$,

$c_1 = 7$, $c_2 = 10$, $n_0 = 1$



$$1/2n^2 - 3n = \Theta(n^2)$$

$$c_1 n^2 \leq 1/2n^2 - 3n \leq c_2 n^2$$

for all $n \geq n_0$. Dividing by n^2 yields

$$c_1 \leq 1/2 - 3/n \leq c_2.$$

$c_1 = 1/14$

$c_2 = 1/2$

$n_0 = 7$

OR

$C_1 = 1 / 4$

$n_0 = 13$

Intuitively, the lower-order terms of an asymptotically positive function can be ignored in determining asymptotically tight bounds because they are insignificant for large n. A tiny fraction of the highest-order term is enough to dominate the lower-order terms. Thus, setting c_1 to a value that is slightly smaller than the coefficient of the highest-order term and setting c_2 to a value that is slightly larger permits the inequalities in the definition of Θ-notation to be satisfied. The coefficient of the highest-order term can likewise be ignored, since it only changes c_1 and c_2 by a constant factor equal to the coefficient.

Example 2

$f(n) = 2n^2 - 5n + 10$

$f(n) \in \Theta(n^2)$??

- $1n^2 \leq 2n^2 - 5n + 10$ … for lower bound
  - $C1 = 1, N_o = 1$
- $2n^2 - 5n + 10 \leq 2n^2$ … for upper bound
  - $C1 = 2, N_o = 2$

$f(n) = (3n^2 / 2) + (5n/2) - 3$

$f(n) \in \Theta(n^2)$??

- $C_1 = 1$
- $C_2 = 2$,
- $N_o = ???$

Example 3

- $f(n) = 3n+3$
- $g(n) = n$
  - $f(n) \in \Theta(n)$

$c_1=2$, $N_o = 1$

$c_2=4$, $N_o = 3$

$N_o = 3$

Suppose …

- $f(n) = 2n^3 - 5n^2 + 10n + 1$
- $g(n) = n^2$

- Is $f(n) \in \Theta(g(n))$???
- No.

$$6n^3 \neq \Theta(n^2)$$

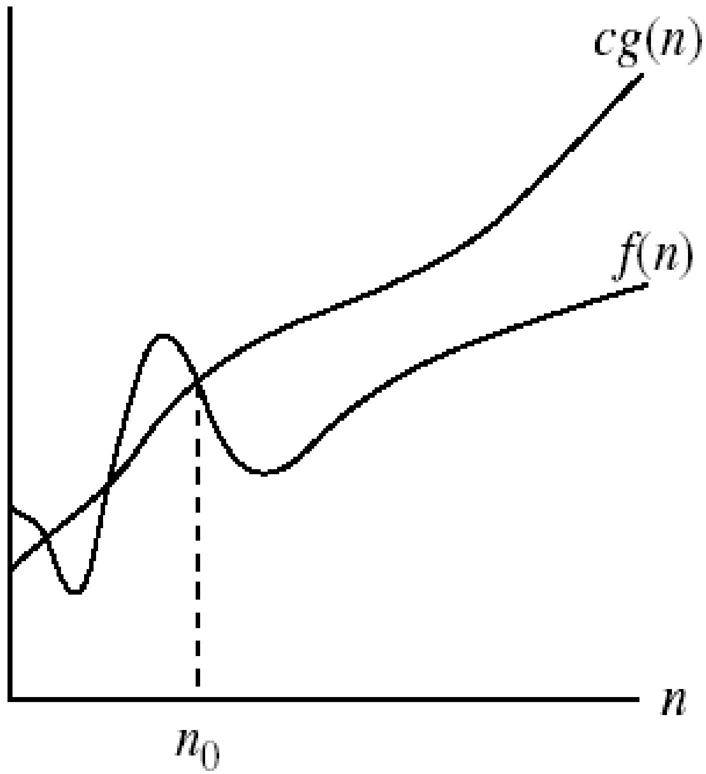
Lower bound: $5n^2 \leq 6n^3$

Upper Bound: $6n^3 \leq ?n^2$ … *always false*

O-notation (Big-O)

- Sometimes we are only interested in proving one bound or the other
- We use O-notation, when we have only an asymptotic upper bound
- O(g(n)) = {f(n) | there exist positive constants 'c' and 'n_0' such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$}
- We write it as $f(n) = O(g(n))$

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



$g(n)$ is an ***asymptotic upper bound*** for $f(n)$.

- If any quadratic function $an^2 + bn + c$ is in $\Theta(n^2)$ is also a $O(n^2)$

- Example:
  - $f(n) = 2n^2$
  - $g(n) = n^2$
  - $f(n) = O(g(n))$

    for $c = 5/2$, $n_0 = 1$

Practice examples

Examples of functions in $O(n^2)$:

n^2

$n^2 + n$

$n^2 + 1000n$

$1000n^2 + 1000n$

Also,

n

$n/1000$

$n^{1.99999}$

$n^2 / \lg \lg \lg n$

- $2n^2 = O(n^3)$:

  $2n^2 \leq cn^3 \Rightarrow$ c = 1 and $n_0 = 2$

- $n^2 = O(n^2)$

  $n^2 \leq cn^2 \Rightarrow$ c = 1 and $n_0 = 1$

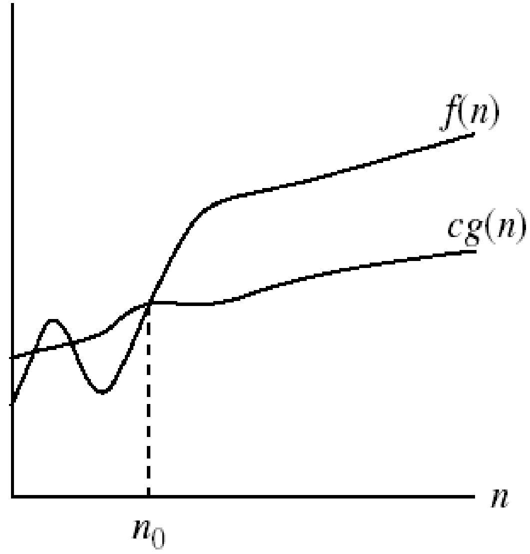
- $1000n^2+1000n = O(n^2)$

  $1000n^2+1000n \leq cn^2 \Rightarrow$ c=1001 and $n_0 = 1000$

Ω - notation

- Just as O-notation provides an asymptotic *upper* bound on a function, omega notation provides an ***asymptotic lower bound.***
  - $\Omega(g(n))$ = the set of functions with a larger or same order of growth as $g(n)$

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



$g(n)$ is an ***asymptotic lower bound*** for $f(n)$.

- $5n^2 = \Omega(n)$
  - $0 \le cn \le 5n^2$, $c = 1$ and $n_0 = 1$
- $100n + 5 \ne \Omega(n^2)$
  - $0 \le cn^2 \le 100n + 5$
- $n = \Omega(n)$
- $n^3 = \Omega(n^2)$

Examples of functions in $\Omega(n^2)$:

n^2

$n^2 + n$

$n^2 - n$

$1000n^2 + 1000n$

$1000n^2 - 1000n$

Also,

n^3

$n^{2.00001}$

$n^2 \lg \lg \lg n$

2^{2^n}

Theorem

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

if and only if

$$f = O(g(n)) \text{ and } f = \Omega(g(n))$$