Asymptotic Notations

Adapted from lectures of

Prof. Charles Leiserson , MIT & CLRC textbook 3nd ed chapter 3

Lecture Outline

- Asymptotic notations
  - *O-, Ω-, and Θ-notation*
  - big-o, big-omega, and big-theta, respectively

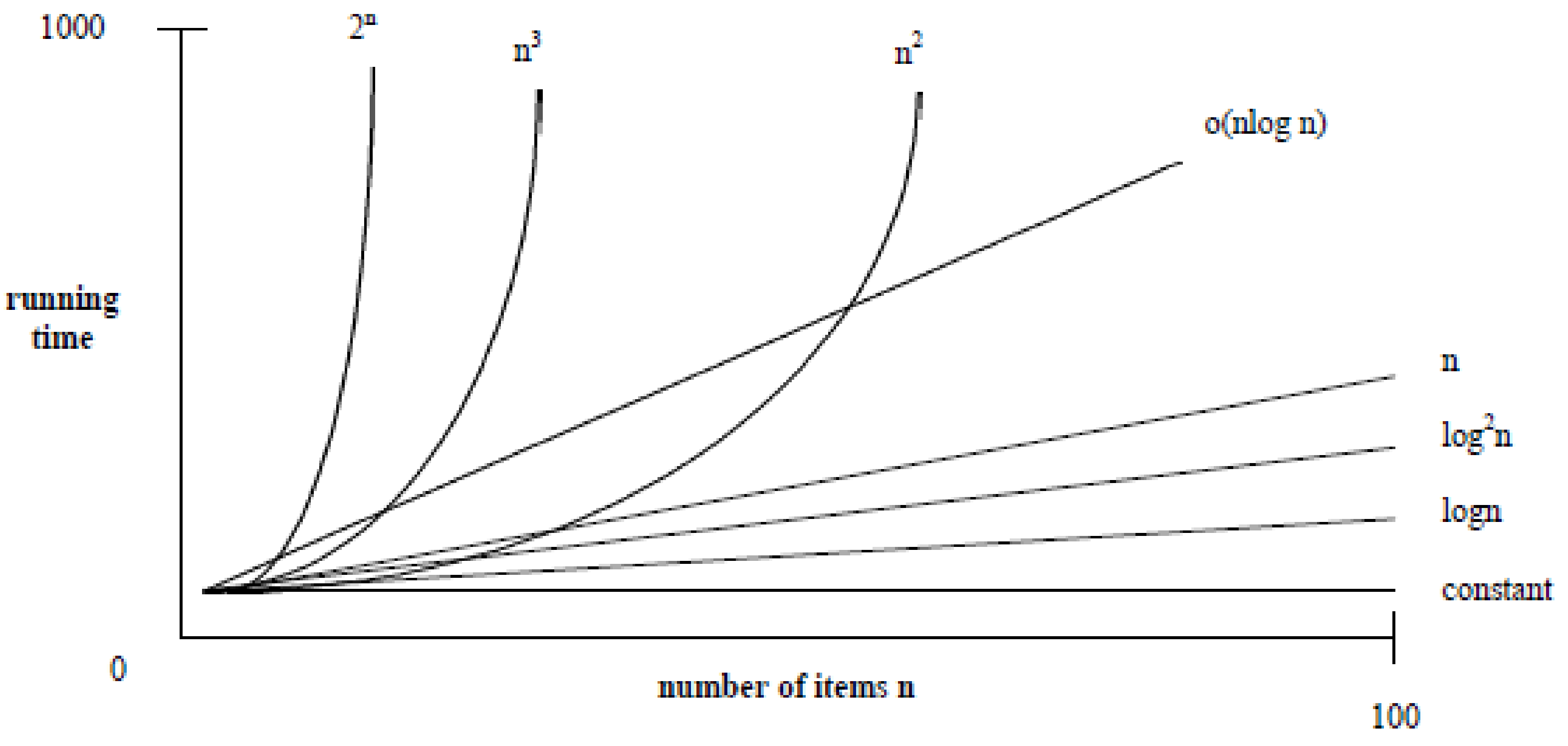
Goal

- The **order of growth** of the RT function of an algorithm gives a simple characterization of the algorithm's efficiency
- It also allows us to compare the relative performance of alternative algorithms.
  - Once the input size *n becomes large enough, merge* sort (**Θ(*n lg n)*** *running time), beats insertion sort (worst-case running* time is **Θ(*n²*)**.*)*
- *It is not always required to find exact RT of an algorithm,* as we did for insertion sort in previous lecture, the extra precision is not usually worth the effort of computing it.
- For large enough inputs, the multiplicative constants and lower-order terms of an exact RT are dominated by the effects of the input size itself.
- Hence, we need a notation to express the growth of the RT function **T(n)** as **n** gets large (asymptotic notations)

Common complexity functions

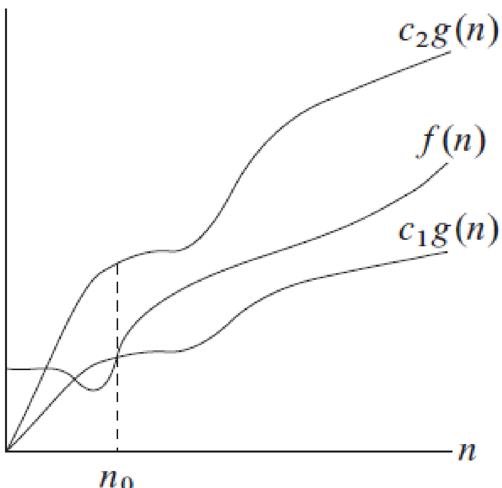
growth rate graph

The growth rates are easily visualized in the following chart. The horizontal axis **x** represent **n** and the vertical **y** axis represent running time:



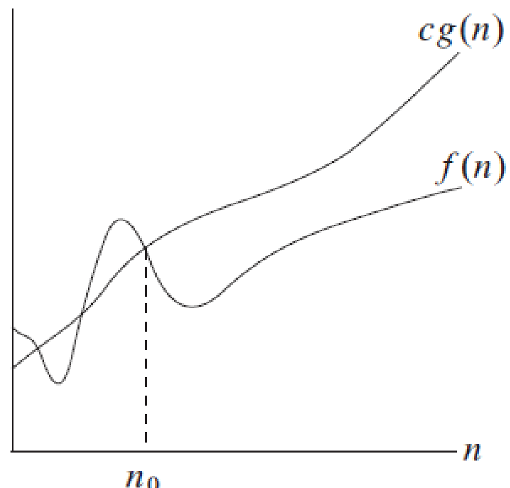
O-, Ω-, and Θ-notation: graphical summary

- f(n): the derived RT of an algorithm
- g(n): one of the known complexity functions



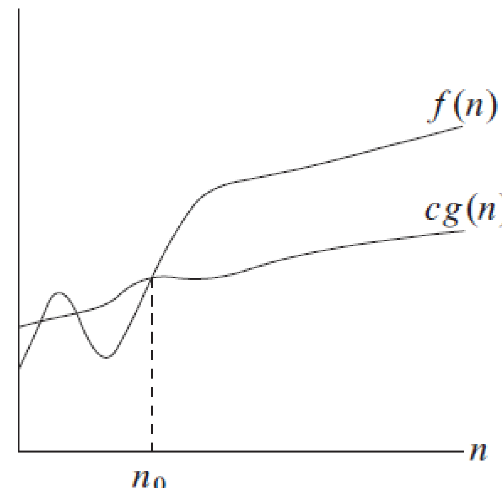
$f(n) = \Theta(g(n))$

(a)



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

(b)



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

(c)

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 $n \ge n_0$.

- {c g(n)} are upper bound of the function f(n)

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

- *functions, not values*

funny, "one-way" equality

NOT Equality !

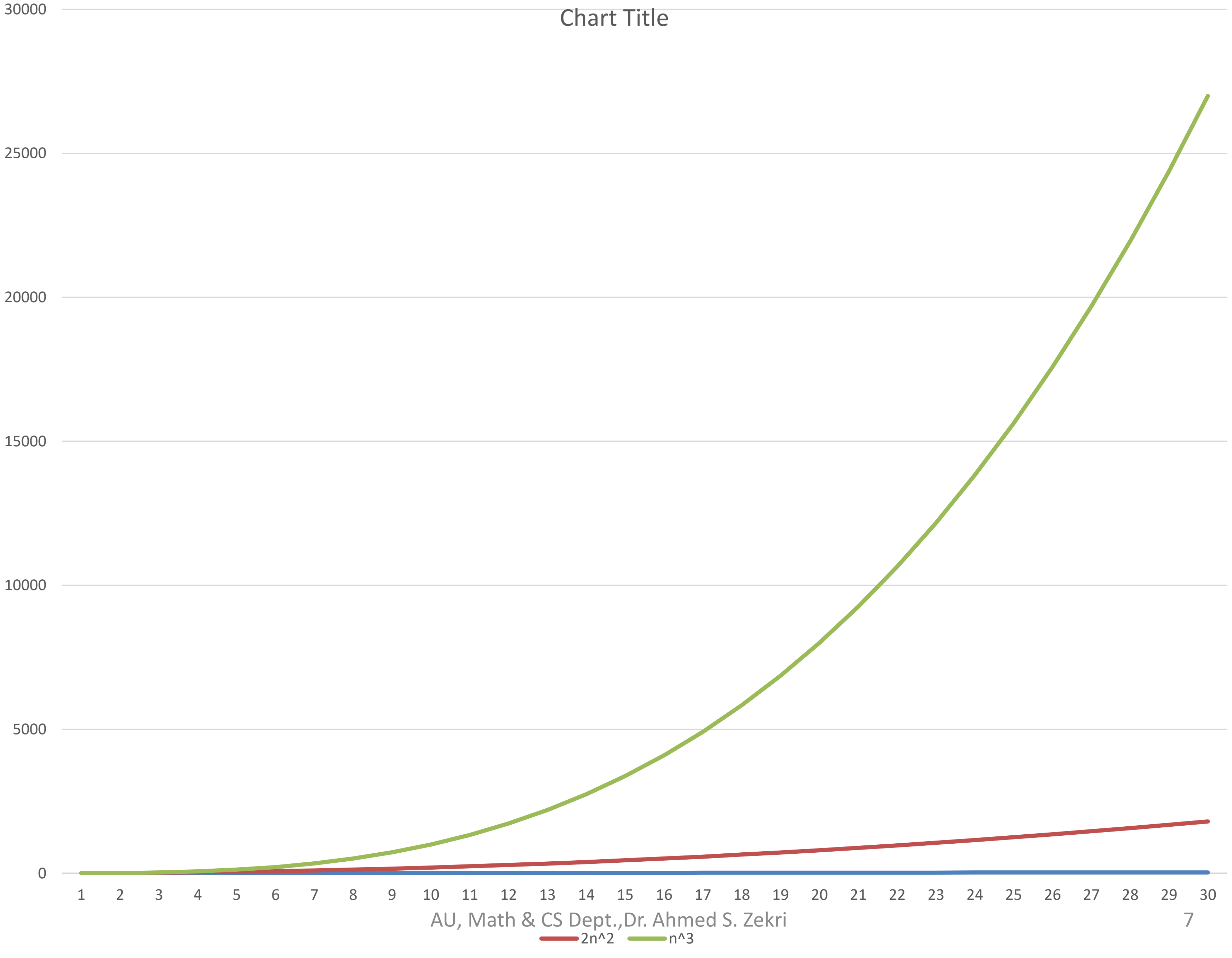


Chart Title
30000
25000
20000
15000
10000
5000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
2n^2
n^3

Set definition of O-notation

- **O(g(n))** is a set of functions {f(n)}

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

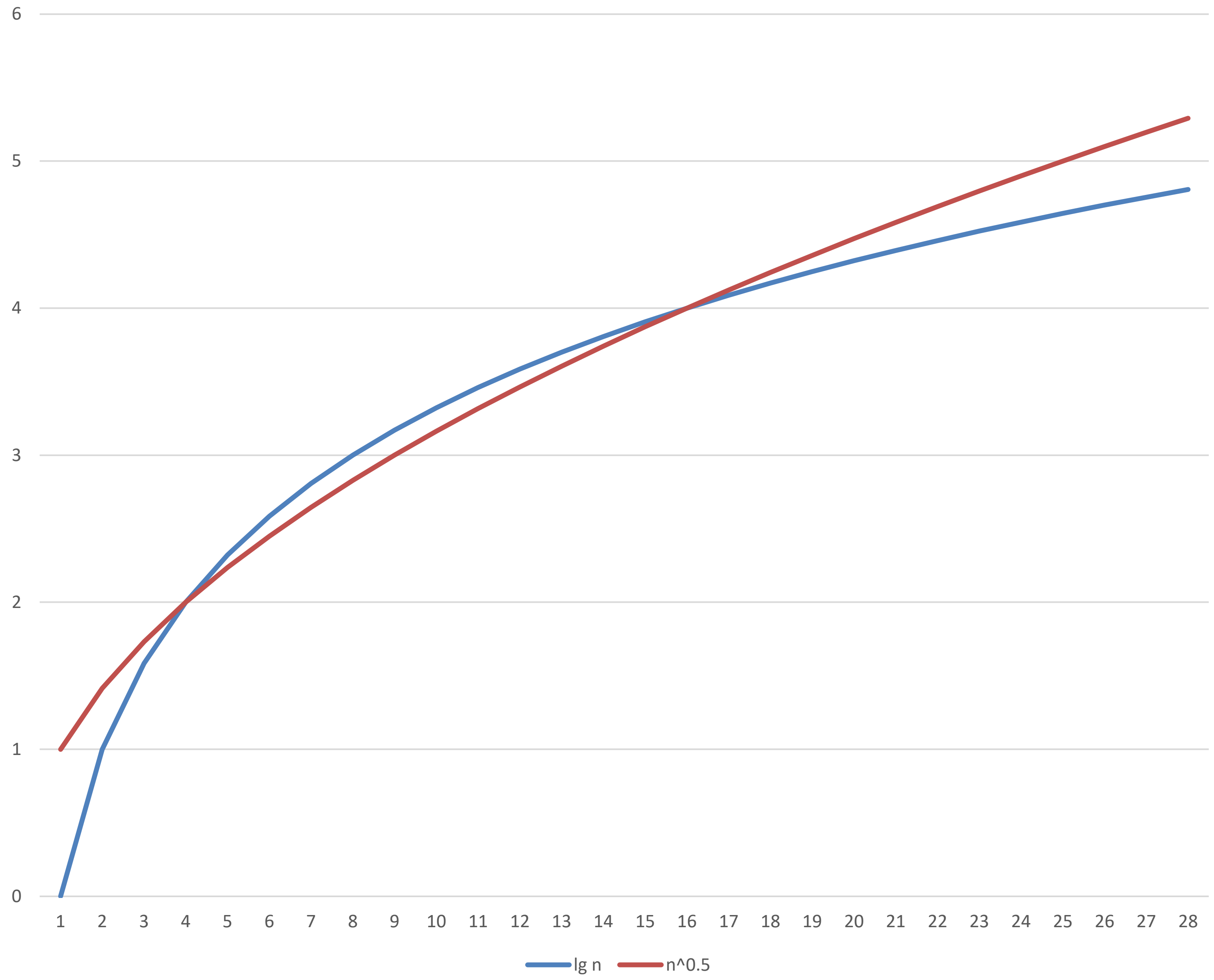
EXAMPLE: $2n^2 \in O(n^3)$

Ω-notation (lower bounds)

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

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

c lg n ≤ √n, *√2=1.4*



6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
lg n
n^0.5

Θ-notation (tight bounds)

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

The Θ-notation asymptotically bounds a function from above and below.

$$\boxed{\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))}$$

EXAMPLE: $\frac{1}{2} n^2 - 3n = \Theta(n^2)$

Example

Show that $\frac{1}{2} n^2 - 3n = \Theta(n^2)$.

- we must determine positive constants c_1, c_2, *and* n_0 *such that*

$$c_1 n^2 \le \tfrac{1}{2} n^2 - 3n \le c_2 n^2$$, for all $n \ge n_0$.

- Dividing by *n yields*

$$c_1 n \le \tfrac{1}{2} n - 3 \le c_2 n$$

Then find c_1 *and* c_2

Example …

Take the RHS to find c2

$\frac{1}{2} n - 3 \le c_2 n$, for all $n \ge n_0$.

- Try to move all terms with n to one side

- $\frac{1}{2} n - c_2 n \le 3$
- $n (\frac{1}{2} - c_2) \le 3$ *multiply by -1*
- $n (c_2 - \frac{1}{2}) \ge -3$
- $n \ge -3/ (c_2 - \frac{1}{2})$
- $n \ge 3/ (\frac{1}{2} - c_2)$

Getting the constants c1, c2, n_0

- We can make the right-hand inequality hold for any value of $n \geq 12$ by choosing any constant $0 < c2 < ½$ (say c2=1/4).
- Likewise, we can make the left-hand inequality hold for any value of $n \geq 8$ by choosing any constant $0 < c1 < ½$ (say c1=1/8).
- Thus, by choosing c1 = 1/8, c2 = 1/4, and $n0 \geq 12$, we can verify that ***$\frac{1}{2} n^2 - 3n = \Theta(n^2)$.***
- Certainly, other choices for the constants c1, c2 exist, but the important thing is that *some* choice exists.

Example

- We can also use the formal definition to verify that $6n^3 \neq \Theta(n^2)$.
- *Suppose for the purpose of* contradiction that *c_2 and n_0 exist such that:*

  $6n^3 \leq c_2 n^2$ *for all $n \geq n0$.*
- *But then* $n \leq c_2/6$ *(divide both sides by n^2) , which* cannot possibly hold for arbitrarily large *n, since c_2 is constant.*

o-notation and ω-notation

- *O-notation and Ω-notation are like ≤and ≥.*
- *o-notation and ω-notation are like < and >.*

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

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

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

EXAMPLE: $n^2/2 = \omega(n)$, $c>2n$