

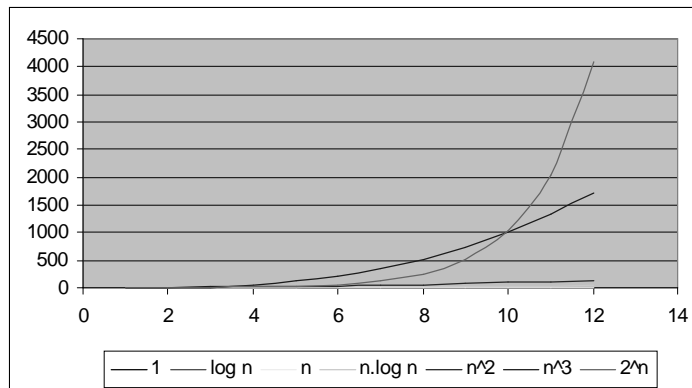
Growth of Functions

How to Compare Functions?

1

Relative Sizes of Functions (move them at the end)

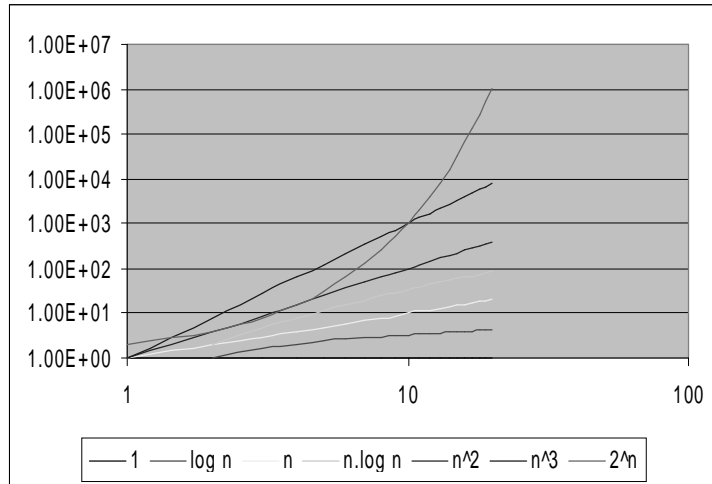
n	1	log n	n	n.log n	n ²	n ³	2 ⁿ
1	1	0	1	0	1	1	2
10	1	3.321928	10	33.21928	100	1000	1024
100	1	6.643856	100	664.3856	10000	1000000	1.26765E+30
1000	1	9.965784	1000	9965.784	1000000	1000000000	1.0715E+301



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-2
ALG00S, javed@kent.edu
Javed I. Khan@1999

Logarithmic Scale



- Order of Functions



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-3
ALG00S, javed@kent.edu
Javed I. Khan@1999

Asymptotics

- Asymptotics means the study of functions of a parameter n , as n becomes larger and larger without bound.
- We will focus on:
 - algorithms when it is running on relatively large data.
 - on major characteristics, without being blinded by the details.
 - On general principles that will apply to the analysis and comparison of many classes of algorithms.



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-4
ALG00S, javed@kent.edu
Javed I. Khan@1999

Simplifying Functions: The Big Picture

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



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-5
ALG00S, javed@kent.edu
Javed I. Khan@1999

Θ -notation

- A function $f(n)$ belongs to the set $\Theta(g(n))$ if there exist positive constants c_1 and c_2 such that it can be sandwiched between $c_1 \cdot g(n)$ and $c_2 \cdot g(n)$, for sufficiently large n .
- Example: $2 \cdot n^2 + 2 = \Theta(n^2)$
- We say $g(n)$ is an **asymptotically tight bound** for $f(n)$.

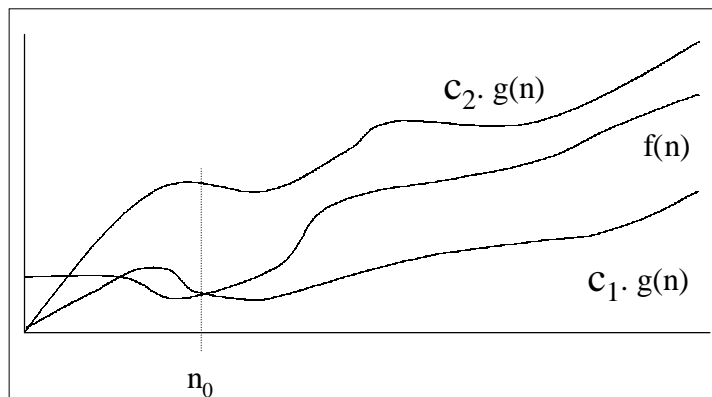
Although it should be “belongs to”, we write it with equality



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-6
ALG00S, javed@kent.edu
Javed I. Khan@1999

Intuitive Picture of Θ -notation



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



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-7
ALG00S, javed@kent.edu
Javed I. Khan@1999

A rule for deriving $g(n)$

- Given any function $f(n)$, throw away the lower order term, and throw away the leading coefficient. It will give the class $g(n)$.
 - $f(n) = \Theta(g(n))$
- Examples:
 - $2 \cdot n^2 - 3n = \Theta(n^2)$
 - $2 \cdot n^5 + 3n = \Theta(n)$



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-8
ALG00S, javed@kent.edu
Javed I. Khan@1999

Proof

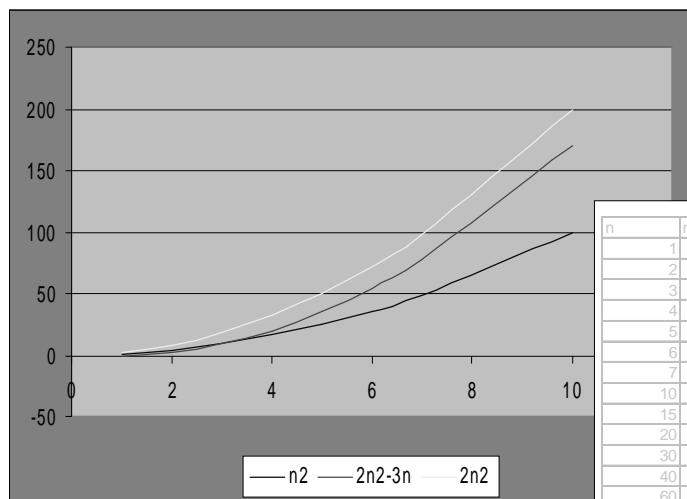
- We must show the positive constants c_1 , c_2 and n_0 such that:
 - $c_1 n^2 \leq 2n^2 - 3n \leq c_2 n^2$
- For all $n \geq n_0$, dividing by n^2 yields:
 - $c_1 \leq 2 - 3/n \leq c_2$
- The R.H.S. in equality can be made to hold:
 - for any $n \geq 1$, by choosing $c_2 \geq 2$
- The L.H.S. in equality can be made to hold:
 - for any $n \geq 3$, by choosing $c_1 \leq 1$



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-9
ALG00S, javed@kent.edu
Javed I. Khan@1999

Excel-Plot



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-10
ALG00S, javed@kent.edu
Javed I. Khan@1999

O-notation (Big-Oh)

- A function $f(n)$ belongs to the set $O(g(n))$ if there exist a positive constants c such that it can be bounded below $c.g(n)$ for sufficiently large n .
- Examples: $2.n^2 + 2 = O(n^2)$
 - If $f(n) = \Theta(g(n))$, it must be $f(n) = O(g(n))$
 - Also, $2.n^2 + 2 = O(n^3)$
- We say $g(n)$ is an **asymptotically upper bound** for $f(n)$.



DESIGN &
ANALYSIS OF
ALGORITHM

Although it should be “belongs to”, we write it with equality

LECT-05, S-11
ALG00S, javed@kent.edu
Javed I. Khan@1999

Ω -notation (Omega)

- A function $f(n)$ belongs to the set $O(g(n))$ if there exist a positive constants c such that it can be bounded above $c.g(n)$ for sufficiently large n .
- Example: $2.n^2 + 2 = \Omega(n^2)$
 - If $f(n) = \Theta(g(n))$, it must be $f(n) = \Omega(g(n))$
 - Also, $2.n^2 + 2 = \Omega(n)$
- We say $g(n)$ is an **asymptotically lower bound** for $f(n)$.



DESIGN &
ANALYSIS OF
ALGORITHM

Although it should be “belongs to”, we write it with equality

LECT-05, S-12
ALG00S, javed@kent.edu
Javed I. Khan@1999

o-notation (little oh)

- O-notation refers to upper bound that is not asymptotically tight.
 - A function $f(n)$ belongs to the set $O(g(n))$ if for all positive constants c , $f(n)$ can be bounded above $c \cdot g(n)$ for sufficiently large n .
 - Or for all c , $0 \leq f(n) \leq c \cdot g(n)$

- Example:2

- $n^2 = O(n^2)$
- $2n^2 = O(n^3)$?
- But:
- $2n^2 \neq o(n^2)$
- $2n^2 = o(n^3)$

$f(n)=o(g(n))$ is a way of saying $f(n)$ is insignificant relative to $g(n)$.



ω -notation (little omega)

- ω -notation refers to lower bound that is not asymptotically tight.
 - A function $f(n)$ belongs to the set $\omega(g(n))$ if for all positive constants c , $f(n)$ can be bounded below $c \cdot g(n)$ for sufficiently large n .
 - Or for all c , $c \cdot g(n) \leq f(n)$

- Example:2

- $n^2 = \Omega(n^2)$
- $2n^2 = \Omega(n)$?
- But:
- $2n^2 \neq \omega(n^2)$
- $2n^2 = \omega(n)$

ω -notation is to Ω -notation, what o -notation is to O -notation.



Comparison of Functions (QUIZ)



DESIGN &
ANALYSIS OF
ALGORITHM

- Transitivity:
- $f(n)=\Theta(g(n))$ and $g(n)=\Theta(h(n))$, $f(n)=\Theta(h(n))$?
- $f(n)=O(g(n))$ and $g(n)=O(h(n))$, $f(n)=O(h(n))$?
- $f(n)=\Omega(g(n))$ and $g(n)=\Omega(h(n))$, $f(n)=\Omega(h(n))$?
- $f(n)=\omega(g(n))$ and $g(n)=\omega(h(n))$, $f(n)=\omega(h(n))$?
- $f(n)=o(g(n))$ and $g(n)=o(h(n))$, $f(n)=o(h(n))$?

LECT-05, S-15
ALG00S, javed@kent.edu
Javed I. Khan@1999

Comparison of Functions (QUIZ)



DESIGN &
ANALYSIS OF
ALGORITHM

- Reflexivity
- Given: $f(n)=\Theta(g(n))$
- $f(n)=O(g(n))$?
- $f(n)=\Omega(g(n))$?
- $f(n)=\omega(g(n))$?
- $f(n)=o(g(n))$?

LECT-05, S-16
ALG00S, javed@kent.edu
Javed I. Khan@1999

Comparison of Functions (QUIZ)

- Symmetry
- Given: $f(n)=\Theta(g(n))$
- $g(n)=\Theta(f(n))$?

- Given: $f(n)=O(g(n))$
- $g(n)=O(f(n))$?
- $g(n)=\Omega(f(n))$?



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-17
ALG00S, javed@kent.edu
Javed I. Khan@1999

Ordering of Function

- We can order a set of function just like a set of integers by using asymptotics:

- if $f(n)=O(g(n))$, $g(n)=O(h(n))$, $h(n)=O(k(n))\dots$
- we can say:
- $f(n)<g(n)<h(n)\dots$



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-18
ALG00S, javed@kent.edu
Javed I. Khan@1999

Comments

- Asymptotics provides us a way of looking at the big picture, without getting lost in the detail.
- It is just one way of looking into the characteristics of computer algorithms, however, knowing an algorithm is much more than its asymptotic analysis.
-
- It does not say anything about the behavior of an algorithm, for small cases.
- Can all function pairs be compared using asymptotics?
 - n and $n^{1 + \sin n}$



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-05, S-19
ALG00S, javed@kent.edu
Javed I. Khan@1999