Μεγάλες διαστάσεις των δεδομένων

Όσο το n γίνεται μεγάλο, τόσο η διαφορά της αποδοτικότητας γίνεται μεγάλη

***

Ασυμπτωτική Πολυπλοκότητα

Πρόβλημα (διάστασης n)

- αλγόριθμος Α πολ/τας 100n
- αλγόριθμος Β πολ/τας n^2

n :

- n=100 Α, Β ίδια αποδοτικότητα
- <100 Β αποδοτικότερος από Α
- >100 Α αποδοτικότερος από Β

Χρόνος στοιχειώδους εντολής =10^{-6}sec

Για n =1000

A εκτελείται σε 10^{-1}sec

B εκτελείται σε 1 s

(B 10 φορές πιο αργά)

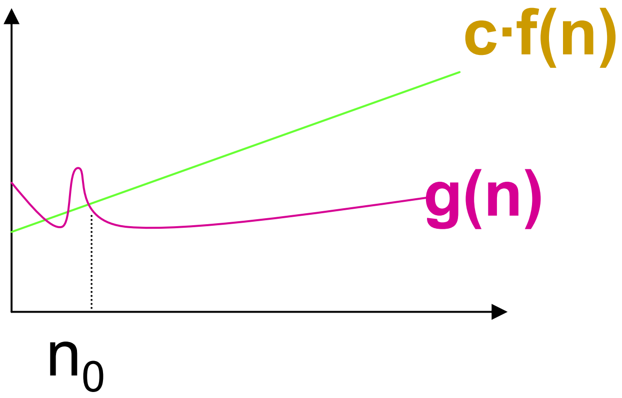
Για n =10000 (B 100 φορές πιο αργός)

Τάξη Μεγέθους

Έστω $f: N \longrightarrow R^+$.

Ορίζουμε τρεις κλάσεις συναρτήσεων σε σχέση με τη συνάρτηση f.

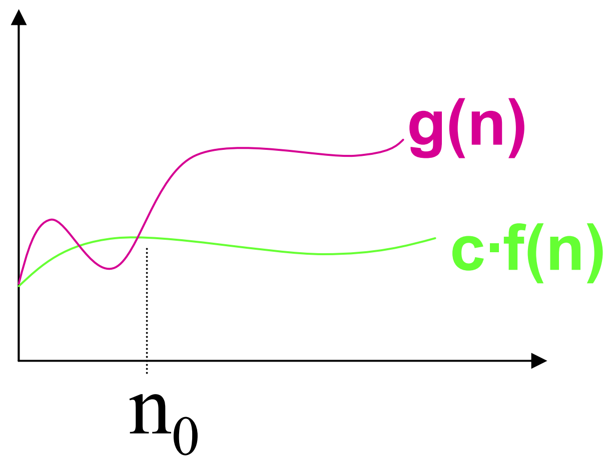
1) $O(f(n))=\{g, \exists n_0 \in N \ \ \exists c \in R^+ \ \ \forall n \geq n_0 \ \ g(n) \leq c \cdot f(n)\}$



Τάξη Μεγέθους

Έστω $f\colon N \longrightarrow R^+$.

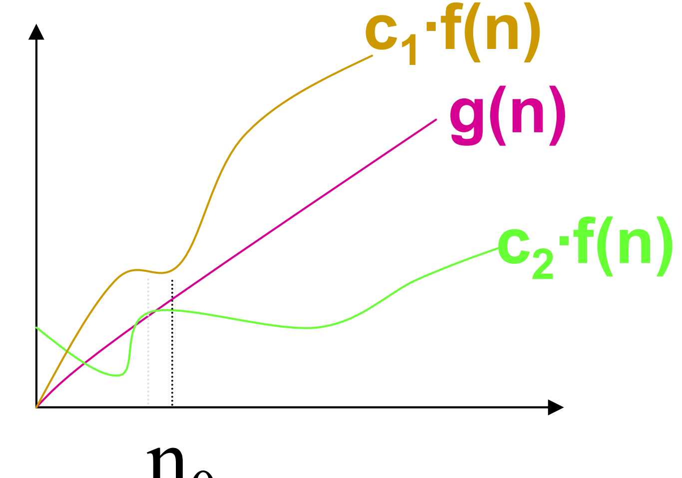
2) $\Omega(f(n))=\{g,\ \exists n_0 \in N\ \ \exists c \in R^+\ \ \forall n \geq n_0\ \ g(n) \geq c\cdot f(n)$



Τάξη Μεγέθους

Έστω f: N $\longrightarrow$ R$^+$.

3) $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$ <u>“ ίδια τάξη με f ”</u>



Τάξη Μεγέθους

Έστω $f: N \longrightarrow R^+$.

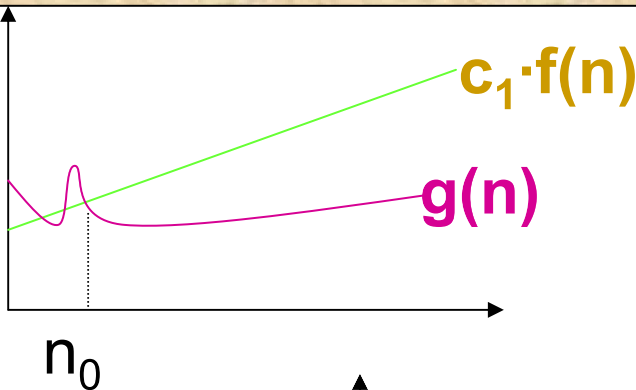
Τρεις κλάσεις συναρτήσεων σε σχέση με τη συνάρτηση f.

$$O(f(n))=\{g,\ \exists n_0 \in N\ \ \exists c \in R^+\ \ \forall n \geq n_0\ \ g(n) \leq c \cdot f(n)\}$$

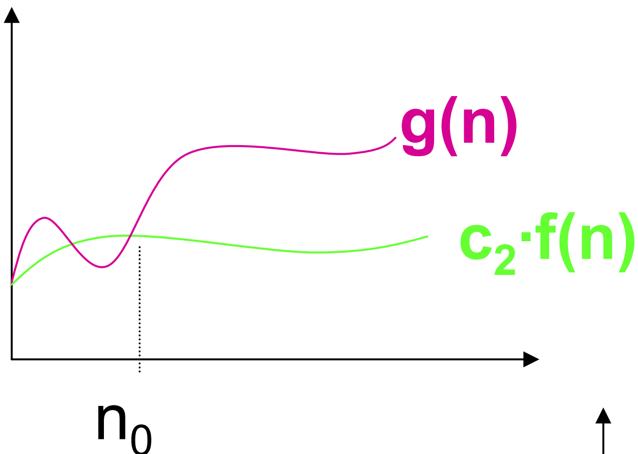
$$\Omega(f(n))=\{g,\ \exists n_0 \in N\ \ \exists c \in R^+\ \ \forall n \geq n_0\ \ g(n) \geq c \cdot f(n)\}$$

$$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

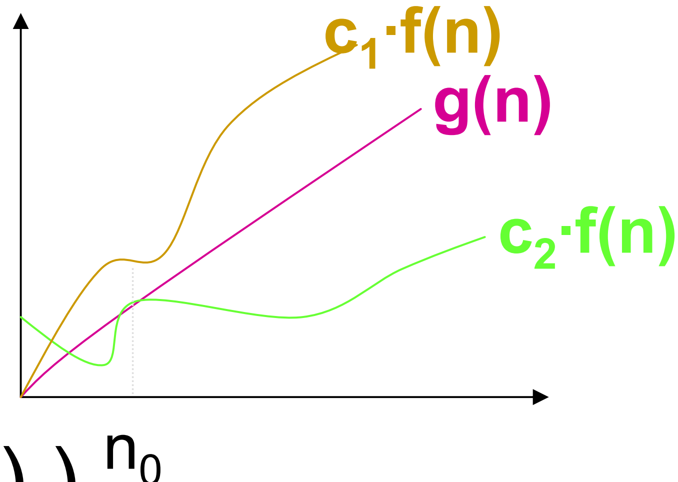
<u>“ ίδια τάξη με f ”</u>



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



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



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

Παραδείγματα:

$3n^2-100n+6=O(n^2)$

διότι $3n^2-100n+6<3n^2$

$3n^2-100n+6=O(n^3)$

διότι $3n^2-100n+6< .00001n^3$

$3n^2-100n+6 \neq O(n)$

διότι $cn<3n^2$ για $n>c$

$f(n)=1$ πολ/τα σταθερή

$f(n)=\log(n)$ πολ/τα λογαριθμική

$f(n)=n$ πολ/τα γραμμική $(an+b)$

$f(n)=n\log(n)$

$f(n)=n^2$

$f(n)=n^3$

$f(n)=n^p$ πολ/τα πολυωνυμική, p σταθερά

$$(a_p n^p + a_{p-1} n^{p-1} + \ldots + a_1 n + a_0)$$

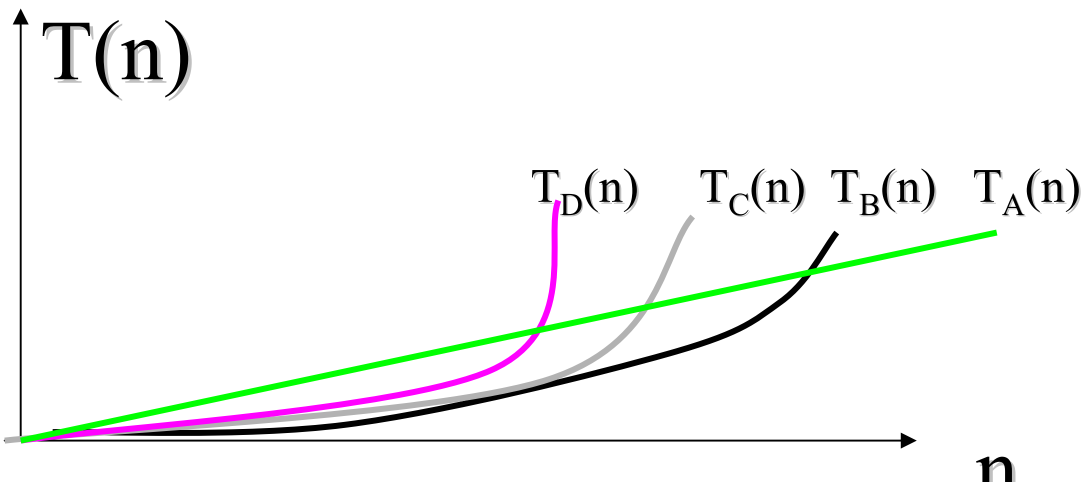
$$\left.\begin{array}{l} f(n)=2^{n} \\ \\ f(n)=a^{n} \end{array}\right\} \text{πολ/τα εκθετική}$$

$f(n)= 2^{2^{n}}$ πολ/τα διπλά εκθετική

Ρυθμός αύξησης του χρόνου εκτέλεσης

Πρόβλημα Π

- <u>A</u> $T_A(n)=O(n)$
- <u>B</u> $T_B(n)=O(n^2)$
- <u>C</u> $T_C(n)=O(n^3)$
- <u>D</u> $T_D(n)=O(2^n)$



Διάσταση n ενός προβλήματος που επιλύουμε σε μια μηχανή 10 MIPS

Complexity Αλγορίθμου	0.1 sec	10 min	1 ημέρα
O(n)	10^6	6×10^9	864×10^9
$O(n^2)$			
$O(n^3)$			
$O(n^4)$			
$O(2^n)$	20	32	39
$O(3^n)$			
O(n!)	10	13	15

Παράδειγμα

Ανάθεση προσωπικού σε εργασίες

1 δυνατότητα $\longrightarrow$ 1μς

20 υπάλληλοι $\longrightarrow$ $2{,}4 \times 10^{18}$

$(T_c > 70.000 \times 10^{6}$ years)

$(T_{Hungarian} = 1{,}95$ s)

Η Αίρεση

Η ανάγκη ανάπτυξης αποτελεσματικών αλγορίθμων θα εκλείψει μπροστά στις αυξητικές αποδόσεις των αυριανών υπολογιστών…

Πρόοδος της ισχύος των υπολογιστών αλλά…

complexity	current size	future size
$O(n)$	n	$n \times 1000$
$O(n^2)$	n	
$O(n^3)$	n	
$O(n^4)$	n	
$O(2^n)$	n	$n+10$
$O(3^n)$	n	
$O(n!)$	n	

Μηχανή 1000 φορές πιο γρήγορη.