# Πώς αξιολογούμε αλγορίθμους

Ο συμβολισμός big-O, O( )

# Απόλυτος Τρόπος Σύγκρισης

- Οι υπολογιστές εξελίσσονται, χρειαζόμαστε «μέτρα και σταθμά» για να μετρήσουμε αλγορίθμους, όχι με απόλυτους χρόνους.
- Κάποιος δοκιμάζει ένα πρόγραμμα σε ένα υπολογιστή και κάποιος άλλος σε άλλον και παίρνουν διαφορετικούς χρόνους.
- Ποιος έχει δίκιο; (μπορεί και οι δύο η κανένας)
- Σύγκριση ανεξάρτητη από Σύστημα (H/W, O/S, Compiler)

# Τα Μέτρα μας

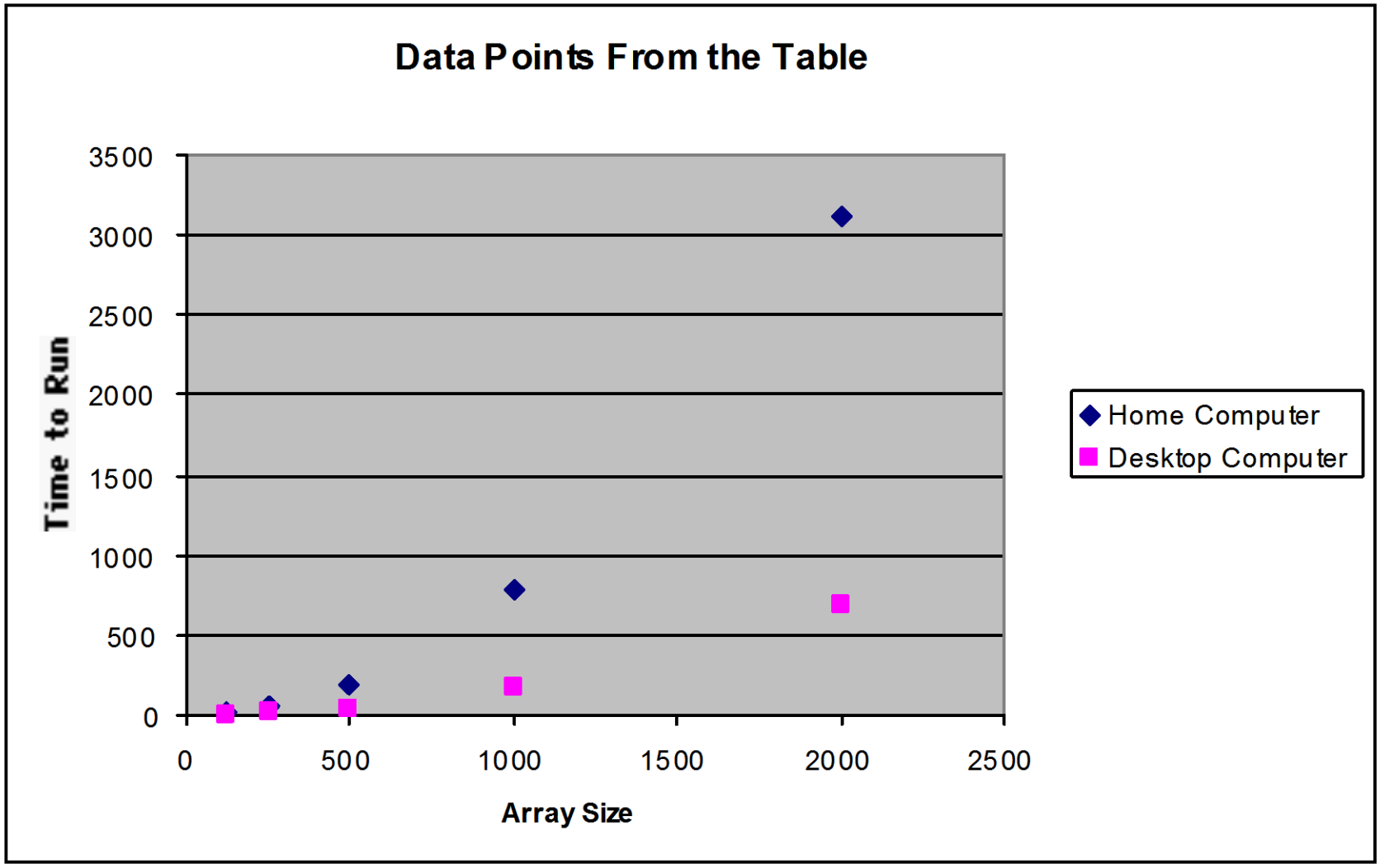
- Εμπειρικά αποτελέσματα απλά συγκρίνουν μέγεθος με χρόνο σε κάποιο μηχάνημα. Χρειαζόμαστε κάτι γενικότερο.
- Το **πλήθος** των δεδομένων. Όσο πιο πολλά δεδομένα τόσο περισσότερος ΧΡΟΝΟΣ και ΧΩΡΟΣ απαιτείται.
- Κριτήριο Χρόνου. Ποια πράξη είναι αυτή που γίνεται πιο συχνά (κυριαρχεί).
  - Απόδοση τιμής,
  - σύγκριση,
  - Πρόσθεση, πολλαπλασιασμός

Παράδειγμα.

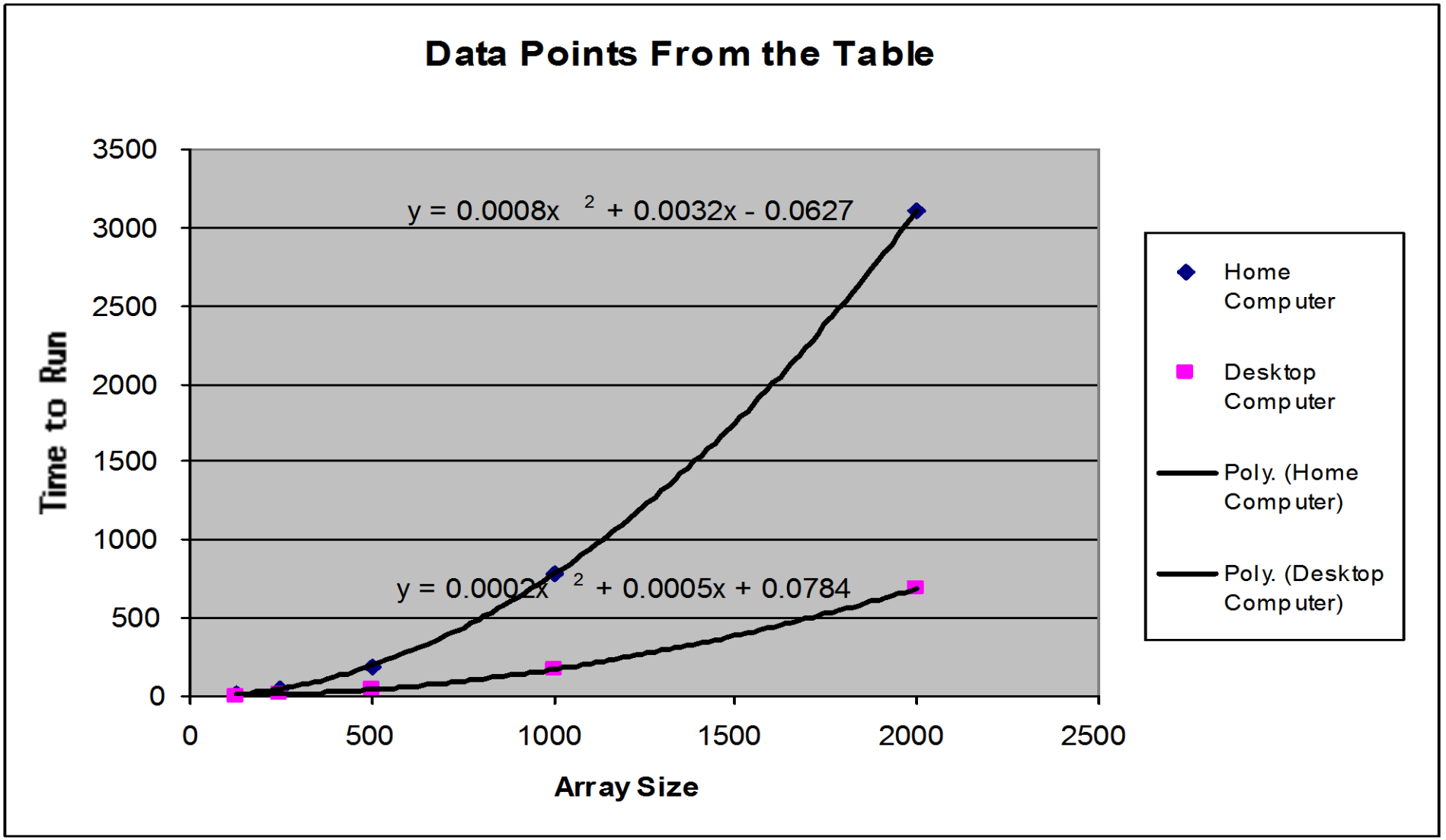
Τρέχουμε ένα απλό αλγόριθμο ταξινόμησης (έστω SelectionSort) σε δύο μηχανές για διάφορα μεγέθη (n):

| Array Size = n | Home Computer | Desktop Computer |
|---|---|---|
| 125 | 12.5 | 2.8 |
| 250 | 49.3 | 11.0 |
| 500 | 195.8 | 43.4 |
| 1000 | 780.3 | 172.4 |
| 2000 | 3114.9 | 690.5 |

# Σημεία μετρήσεων



# υπολογίζουμε εξισώσεις-τάσεις



# Ο συμβολισμός big-O O( )

Έχουμε δύο περιπτώσεις

$F_1(n) = 0.0008\ n^2 + 0.0032\ n + 0.0627$

$F_2(n) = 0.0002\ n^2 + 0.0005\ n + 0.0784$

το $n^2$ μεγαλώνει πιο γρήγορα από το n, δεν περιμένουμε οι συναρτήσεις να μεγαλώνουν πολύ χειρότερα από $0.0008\ n^2$ and $0.0002\ n^2$, αντίστοιχα. Μπορούμε να βρούμε σταθερές

$C_1 > 0.0008$ και $C_2 > 0.0002$

ώστε

$F_1(n) <= \mathbf{C_1}\ n^2$ και $F_2(n) <= \mathbf{C_2}\ n^2$ για "μεγάλα" n.

# $O(n^2)$

Λέμε ότι $F_1(n) = O(n^2)$, και επίσης $F_2(n) = O(n^2)$.

Το νόημα του συμβολισμού είναι ότι υπάρχουν σταθερές **$C_1$** και **$C_2$**, καθώς και θετικοί ακέραιοι **$N_1$** and **$N_2$** ώστε

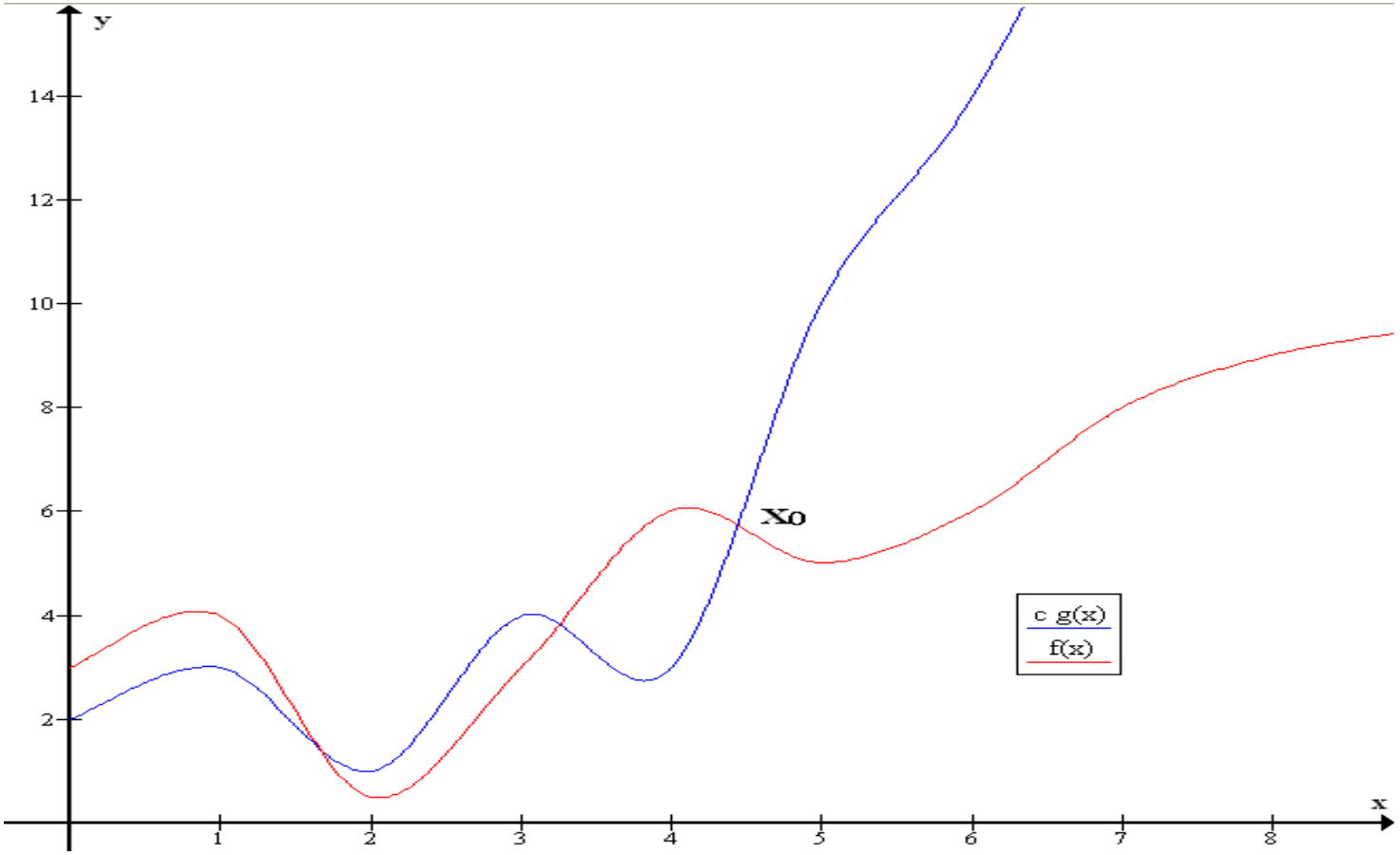
$F_1(n) <= C_1\ n^2$ για όλα τα $n >= N_1$

και

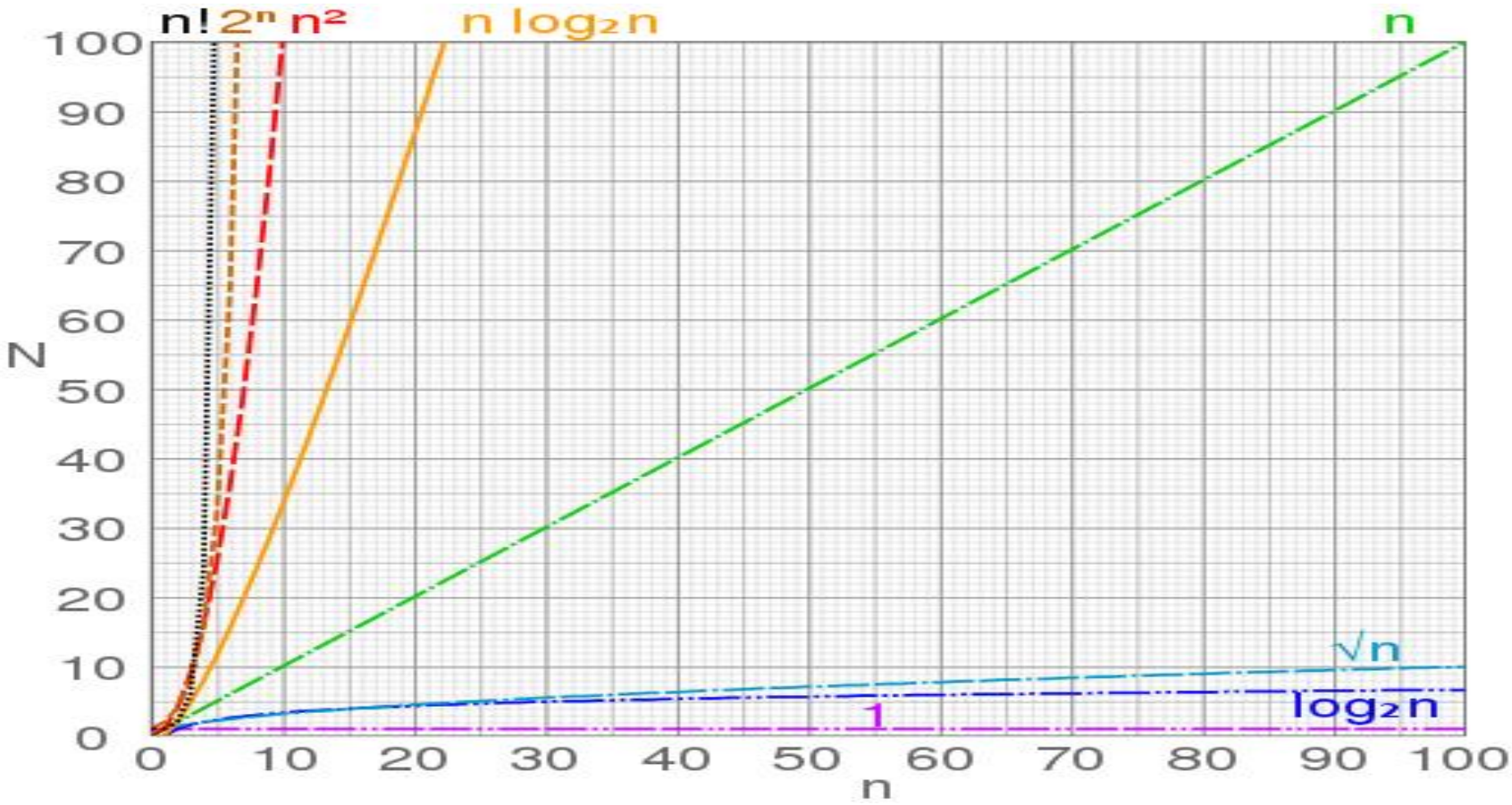
$F_2(n) <= C_2\ n^2$ για όλα τα $n >= N_2$.

Δεν διαφοροποιούμε την απόδοση της F1 και F2. Μας ενδιαφέρει η συμπεριφορά του αλγόριθμου και όχι ο ακριβής χρόνος τρεξίματος σε κάποια μηχανή.

# Σταθερές c και N



# 1, logn, √n, n, nlogn, $n^2$, $2^n$, n!



# Μερικές Συγκρίσεις: F(n)

| n | $\log_{10}(n)$ | $n^{1/2}$ | $n \log_{10}(n)$ | $n^2$ | $n^3$ | $2^n$ | $n^n$ |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 2 | 1 |
| 10 | 1 | $10^{0.5}$ | 10 | $10^2$ | $10^3$ | 1024 | $10^{10}$ |
| $10^2$ | 2 | 10 | $2*10^2$ | $10^4$ | $10^6$ | $\approx 10^{30}$ | $10^{200}$ |
| $10^3$ | 3 | $10^{1.5}$ | $3*10^3$ | $10^6$ | $10^9$ | $\approx 10^{300}$ | |
| $10^4$ | 4 | $10^2$ | $4*10^4$ | $10^8$ | $10^{12}$ | | |
| $10^5$ | 5 | $10^{2.5}$ | $5*10^5$ | $10^{10}$ | $10^{15}$ | | |
| $10^6$ | 6 | $10^3$ | $6*10^6$ | $10^{12}$ | $10^{18}$ | | |
| $10^7$ | 7 | $10^{3.5}$ | $7*10^7$ | $10^{14}$ | $10^{21}$ | | |
| $10^8$ | 8 | $10^4$ | $8*10^8$ | $10^{16}$ | $10^{24}$ | | |
| $10^9$ | 9 | $10^{4.5}$ | $9*10^9$ | $10^{18}$ | $10^{27}$ | | |
| $10^{10}$ | 10 | $10^5$ | $10*10^{10}$ | $10^{20}$ | $10^{30}$ | | |

Παρατήρηση: $3.15*10^{13}$ microseconds σε ένα έτος.

Ο() χρήσιμος για πρόβλεψη χρόνου που απαιτείται και καθορίζει στρατηγικές σχεδιασμού αλγορίθμων.

Αντικείμενο μελλοντικού μαθήματος

(Σχεδιασμός Αλγορίθμων)