Ασυμπτωτική Ανάλυση και Ασυμπτωτικός Συμβολισμός

Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

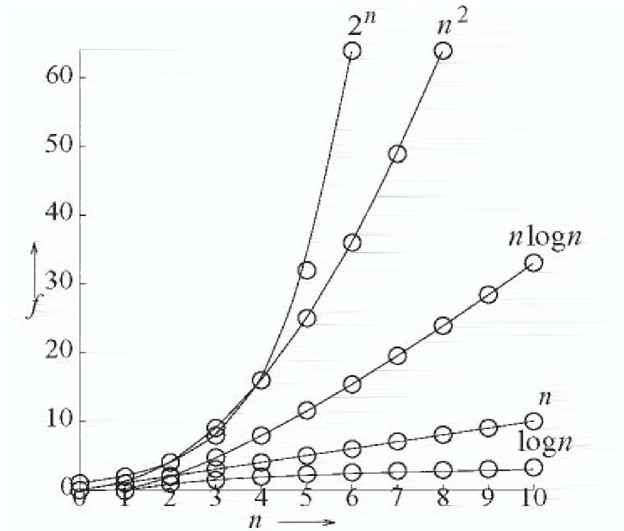
Πανεπιστήμιο Αιγαίου

Ασυμπτωτική Εκτίμηση

- ... αγνοεί **σταθερές** και εστιάζει σε **τάξη μεγέθους** χρόνου εκτέλεσης **συναρτήσει του** n.
- Σταθερές εξαρτώνται από υπολογιστή, υλοποίηση, κλπ.
  Παράδειγμα αλγόριθμος max με $3n - 1$ λειτουργίες.
  Υπολογιστής με 10 λειτ/msec, χρόνος $\frac{3}{10}n - \frac{1}{10}$ msecs.
  Υπολογιστής με 100 λειτ/msec, χρόνος $\frac{3}{100}n - \frac{1}{100}$ msecs.
- Τάξη μεγέθους είναι **εγγενής ιδιότητα** του αλγόριθμου.
  max έχει **γραμμικό χρόνο** σε **όλους** τους υπολογιστές.
- Εστιάζουμε σε (πολύ) **μεγάλα στιγμιότυπα**.
  Καλύτερος αλγόριθμος ⇔ χ.ε. **μικρότερης τάξης μεγέθους**.

Ασυμπτωτική Εκτίμηση

Σταθερό $\sim 1 \leq$ **λογαριθμικό** $\sim \log n \leq$ **γραμμικό** $\sim n \leq \sim n \log n$
$\leq$ **τετραγωνικό** $\sim n^2 \leq$ **κυβικό** $\sim n^3 \leq$ **εκθετικό** $\sim 2^n$

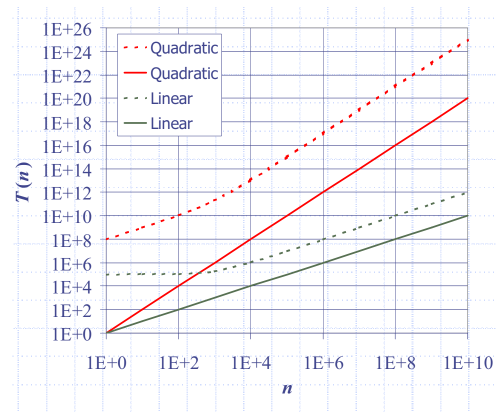


Σταθερές;

- **Τάξη μεγέθους** δεν εξαρτάται από **σταθερές**!
  δεν καθορίζεται από **όρους μικρότερης τάξης**!
- Αγνοούμε **εντελώς** κάθε **σταθερά** και **όρο μικρότερης τάξης**.
  Κρατάμε μόνο **κυρίαρχο όρο**.

$10^2 n + 10^5$ είναι **γραμμικό**,
δηλ. έχει *τάξη μεγέθους* n.

$10^5 n^2 + 10^8 n$ είναι **τετραγωνικό**,
δηλ. έχει *τάξη μεγέθους* n^2.



Ασυμπτωτικός Συμβολισμός

- …εκφράζει **αποτελέσματα** ασυμπτωτικής εκτίμησης.
- $\Theta(\cdot)$ δηλώνει **ακριβή εκτίμηση** της τάξης μεγέθους.
  $f(n) = \Theta(g(n))$ ανν υπάρχουν θετικές σταθερές c_1, c_2 και n_0:
  $$\forall n \geq n_0\,,\quad c_1\,g(n) \leq f(n) \leq c_2\,g(n)$$
  $an^2 + bn + c = \Theta(n^2)$, $500n^2 + 100n^3 + 10^{-5}n^3 \log n = \Theta(n^3 \log n)$.
- $\mathrm{O}(\cdot)$ δηλώνει **άνω φράγμα** στην τάξη μεγέθους.
  $f(n) = \mathrm{O}(g(n))$ ανν υπάρχουν θετικές σταθερές c και n_0:
  $$\forall n \geq n_0\,,\quad f(n) \leq c\,g(n)$$
  $100n^3 + 10^{-5}n^3 \log n = \mathrm{O}(n^3 \log n) = \mathrm{O}(n^4)$ αλλά $10^{-10}n^2 \neq \mathrm{O}(n)$.
- $\Omega(\cdot)$ δηλώνει **κάτω φράγμα** στην τάξη μεγέθους.
  $f(n) = \Omega(g(n))$ ανν υπάρχουν θετικές σταθερές c και n_0:
  $$\forall n \geq n_0\,,\quad f(n) \geq c\,g(n)$$
  $10^{-5}n^3 \log n = \Omega(n^3 \log n) = \Omega(n^3)$ αλλά $10^{10}n \neq \Omega(n^2)$.

Ασυμπτωτικός Συμβολισμός

- $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = \mathrm{O}(g(n))$ και $f(n) = \Omega(g(n))$.
- $\mathrm{o}(\cdot)$ δηλώνει **άνω φράγμα** που δεν είναι ακριβές.
  $f(n) = \mathrm{o}(g(n))$ ανν για κάθε σταθερά $c > 0$, υπάρχει σταθερά n_0:
  $$\forall n \geq n_0\,,\quad f(n) < c\,g(n) \quad \text{ή} \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$
  $5n^3 \log n = \mathrm{o}(n^4)$ αλλά $10n^2 \neq \mathrm{o}(n^2)$.
- $\omega(\cdot)$ δηλώνει **κάτω φράγμα** που δεν είναι ακριβές.
  $f(n) = \omega(g(n))$ ανν για κάθε σταθερά $c > 0$, υπάρχει σταθερά n_0:
  $$\forall n \geq n_0\,,\quad f(n) > c\,g(n) \quad \text{ή} \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$
  $5n^3 \log n = \omega(n^3)$ αλλά $10n^2 \neq \omega(n^2)$.

Ασυμπτωτικός Συμβολισμός

- $f(n) = \Theta(g(n)) \sim$ ασυμπτωτικά $f(n) = g(n)$
  $f(n) = \mathrm{O}(g(n)) \sim$ ασυμπτωτικά $f(n) \leq g(n)$
  $f(n) = \Omega(g(n)) \sim$ ασυμπτωτικά $f(n) \geq g(n)$
  $f(n) = \mathrm{o}(g(n)) \sim$ ασυμπτωτικά $f(n) < g(n)$
  $f(n) = \omega(g(n)) \sim$ ασυμπτωτικά $f(n) > g(n)$.
- Κρατάμε μόνο τον **κυρίαρχο όρο**.
- Πολυώνυμο βαθμού d: $a_d n^d + a_{d-1}n^{d-1} + a_1 n + a_0 = \Theta(n^d)$.
- $\sum_{i=1}^n i = \Theta(n^2)$, $\sum_{i=1}^n i^2 = \Theta(n^3)$, $\ldots \sum_{i=1}^n i^k = \Theta(n^{k+1})$.
- $\sum_{i=1}^n 1/i = \Theta(\log n)$, $\sum_{i=1}^n 2^i = \Theta(2^n)$, $\log(n!) = \Theta(n \log n)$.

Ασκήσεις

Αληθείς ή **ψευδείς** και γιατί;

1. $10f(n) + 10^{10} = \mathrm{O}(f(n))$. **Α**
2. $f(n) + g(n) = \Theta(\min\{f(n), g(n)\})$. **Ψ**
3. $f(n) + g(n) = \Omega(\min\{f(n), g(n)\})$. **Α**
4. $f(n) + g(n) = \mathrm{O}(\max\{f(n), g(n)\})$. **Α**

Ασκήσεις

Να τοποθετηθούν οι συναρτήσεις σε **αύξουσα σειρά** τάξης μεγέθους:

2^{5n}	$\log^4 n$	$(\log n)^{100} \log\log n$	$n \log\log n$	$n^{0.1} \log\log n$
2^n	$n^{0.6}$	$2^n + n^{2^{100}}$	$n^{1/\log n}$	$\log(n!)$
$n^{\log n}$	$\log\log n$	$2^{\log^3 n}$	$\frac{n}{\log_n 2} + n$	$(\log n)^{\log n}$

Πότε $f(n)$ **είναι** $\oplus(g(n))$ και πότε **δεν είναι** ($\oplus \in \{\Theta, \mathrm{O}, \Omega, \mathrm{o}, \omega\}$);

$f(n)$	$g(n)$	$\Theta(g(n))$	$\mathrm{O}(g(n))$	$\mathrm{o}(g(n))$	$\Omega(g(n))$	$\omega(g(n))$
2^{n+5}	$2^n + 2^5 + n^{100}$					
$n^4 - n^3$	$16^{\log n}$					
5^{4n}	10^{2n}					
$n^{1/\log\log n}$	$n^{0.001}$					
$n!$	n^n					
$n^{\log^{20} n}$	2^n					

Ασυμπτωτική Ανάλυση

- Ασυμπτωτική εκτίμηση χρόνου εκτέλεσης ενός αλγορίθμου στην **χειρότερη**, **μέση**, και **καλύτερη** περίπτωση.
- Μετράμε αριθμό **βημάτων** (όχι **λειτουργίες**).
  Βήμα: σύνολο εντολών με χ.ε. **ανεξάρτητο από μέγεθος στιγμιότυπου**.
- Επιλέγουμε βήματα που **καθορίζουν** ασυμπτωτική συμπεριφορά χρόνου εκτέλεσης.

Άθροισμα Προθεμάτων (Prefix Sums)

Είσοδος: $A[n]$ (π.χ. $[2, 5, 3, 10]$)
Έξοδος: $P[n]$, $P[j] = \sum_{i=0}^{j} A[i]$, $j = 0, \ldots, n-1$ (π.χ. $[2, 7, 10, 20]$)

```
int *slowPrefixSums(int A[ ], int n)
    P ← new int[n];
    for j ← 0 to n − 1 do
        P[j] ← A[0];
        for i ← 1 to j do
(*)         P[j] ← P[j] + A[i];
    return(P);
```

Το (*) εκτελείται j φορές για κάθε $j = 0, \ldots, n-1$.
Συνολικά, $\sum_{j=0}^{n-1} j = \Theta(n^2)$.
Χρόνος εκτέλεσης: $\Theta(n^2)$
(σε **κάθε περίπτωση**).

```
int *prefixSums(A[ ], int n)
    P ← new int[n];
    P[0] ← A[0];
    for j ← 1 to n − 1 do
(*)     P[j] ← P[j − 1] + A[j];
    return(P);
```

Το (*) εκτελείται $n-1$ φορές.
Χρόνος εκτέλεσης prefixSums: $\Theta(n)$
(σε **κάθε περίπτωση**).

Χειρότερη - Μέση - Καλύτερη Περίπτωση

```
int linearSearch(A[ ], int n, int x)
    for i ← 0 to n − 1 do
(*)     if A[i] = x then return(i);
    return(−1);
```

Αποτυχημένη αναζήτηση:
(*) εκτελείται n φορές.
Χρόνος εκτέλεσης: $\Theta(n)$
(σε **κάθε** περίπτωση).

Επιτυχημένη αναζήτηση:

- **Καλύτερη περίπτωση**: $A[0] = x$, χρόνος $\Omega(1)$.
- **Χειρότερη περίπτωση**: $A[n-1] = x$, χρόνος $\mathrm{O}(n)$.
- **Μέση περίπτωση**: υποθέτουμε **κατανομή** $\mathbb{P}[A[i] = x] = 1/n$.
  Μέση τιμή χρόνου $\sum_{i=1}^{n} \frac{i}{n} = \Theta(n)$.
- Ενδεχόμενα του χ.ε. για **συγκεκριμένο** (μέγεθος) στιγμιότυπου n.
  Λάθος: Καλύτερη περίπτωση όταν n μικρό, χειρότερη όταν n μεγάλο!

Πρακτικά Αποδοτικοί Αλγόριθμοι

- … έχουν **πολυωνυμική** πολυπλοκότητα (π.χ. $\log n$, n, $n \log n$, n^2, n^3).
  Σπάνιοι αλγόριθμοι με πολυπλοκότητα n^d, όπου d μεγάλος αριθμός.
- Μεγαλύτερη (**εκθετική**) πολυπλοκότητα **απαγορευτική** για μεγάλα στιγμιότυπα! Π.χ. $100n^2 < 2^{n/5}$ για κάθε $n \geq 100$.
- Πόσο **μεγαλώνουν** τα μεγέθη που λύνουμε (σε συγκεκριμένο χρόνο) όταν **10πλασιάζεται** η ταχύτητα.

Πολυπλ.	n πριν	n' μετά	Λόγος
$100 \log n$	2^{100}	2^{1000}	2^{900}
$10n$	1000	10000	10
$1000n$	10	100	10
$10n \log n$	140	1003	7.16
$5n^2$	44	141	$\sqrt{10} = 3.16$
2^n	13	16	1.25 $(n' = n + \log 10)$