

Ασυμπτωτικός Συμβολισμός

Δημήτρης Φωτάκης

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο



Υπολογιστική Πολυπλοκότητα

- Υπολογιστική πολυπλοκότητα **αλγόριθμου A**:
 - Ποσότητα υπολογιστικών πόρων που απαιτεί A ως **αύξουσα συνάρτηση μεγέθους** στιγμιότυπου εισόδου.
 - Χρόνος, μνήμη, επεξεργαστές, επικοινωνία, τυχαιότητα.
 - **Χειρότερης, μέσης, καλύτερης** περίπτωσης.
- Μέγεθος στιγμιότυπου εισόδου **n** :
 - #bits για αναπαράσταση **δεδομένων εισόδου** στη μνήμη.
 - Πλήθος **βασικών συνιστωσών** που αποτελούν μέτρο μεγέθους και δυσκολίας στιγμιότυπου (π.χ. κορυφές & ακμές γράφου).
- Υπολογιστική πολυπλοκότητα **προβλήματος Π**:
 - Πολυπλοκότητα (χειρότερης περίπτωσης) καλύτερου αλγόριθμου που λύνει πρόβλημα Π.

Ανάλυση Αλγορίθμου

- Απόδειξη ορθότητας
 - Μερικές φορές για ένα καλώς ορισμένο **υποσύνολο** των στιγμιοτύπων εισόδου.
- Εκτίμηση υπολογιστικής πολυπλοκότητας.
 - **Χειρότερης, μέσης, και καλύτερης** περίπτωσης.
- Καταλληλότερη λύση ανάλογα με **απαιτήσεις εφαρμογής.**

Ασυμπτωτική Εκτίμηση

- Χρόνος εκτέλεσης αλγόριθμου A :
 - Αύξουσα συνάρτηση του $T(n)$ που εκφράζει σε πόσο χρόνο ολοκληρώνεται ο A όταν εφαρμόζεται σε στιγμ. μεγέθους n .
- Ενδιαφέρει η τάξη μεγέθους $T(n)$ και όχι ακριβής εκτίμηση $T(n)$.
 - Ακριβής εκτίμηση είναι συχνά δύσκολη και εξαρτάται από υπολογιστικό περιβάλλον, υλοποίηση, ...
 - Τάξη μεγέθους είναι εγγενής ιδιότητα του αλγόριθμου.
 - Δυαδική αναζήτηση έχει λογαριθμικό χρόνο.
 - Γραμμική αναζήτηση έχει γραμμικό χρόνο.
- **Ασυμπτωτική εκτίμηση** αγνοεί σταθερές και εστιάζει σε τάξη μεγέθους χρόνου εκτέλεσης.

Ασυμπτωτικός Συμβολισμός

□ ... εκφράζει τα **αποτελέσματα** ασυμπτωτικής εκτίμησης.

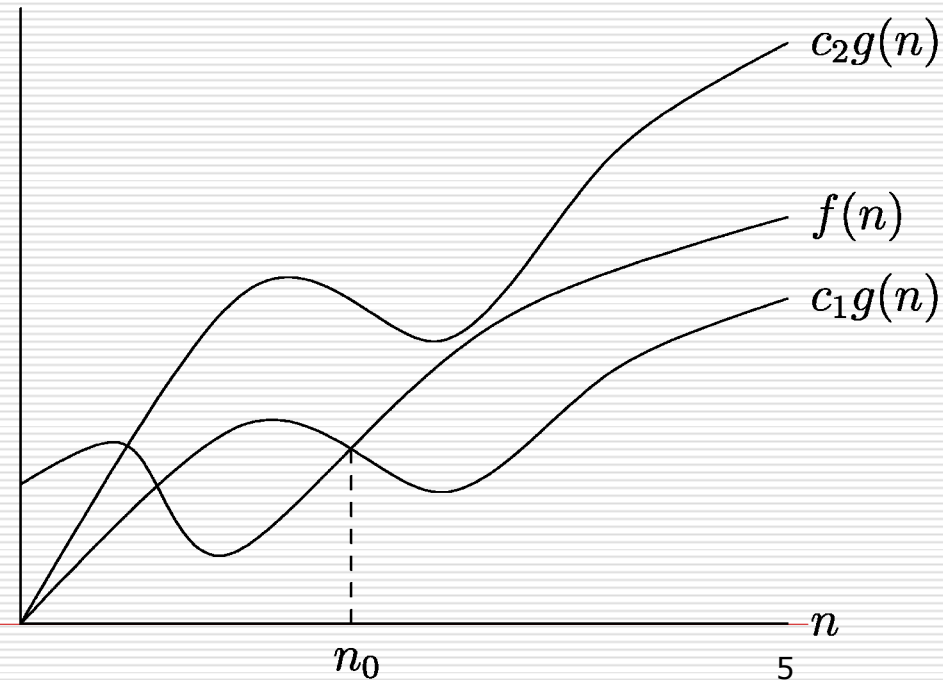
□ $\Theta(\)$ δηλώνει την **ακριβή εκτίμηση** τάξης μεγέθους.

$f(n) \in \Theta(g(n))$ ή $f(n) = \Theta(g(n))$ αν \exists σταθερές $c_1, c_2, n_0 > 0$:
 $\forall n \geq n_0, c_1 g(n) \leq f(n) \leq c_2 g(n)$

■ $\Theta(g(n))$ **σύνολο** συναρτήσεων **ίδιας** τάξης μεγέθους με $g(n)$.

$$an^2 + bn + c = \Theta(n^2)$$

$$500n^2 + 100n^3 + 10^{-5}n^3 \log n = \Theta(n^3 \log n)$$



Ασυμπτωτικός Συμβολισμός O

□ $O(\)$ δηλώνει **άνω φράγμα** στην τάξη μεγέθους.

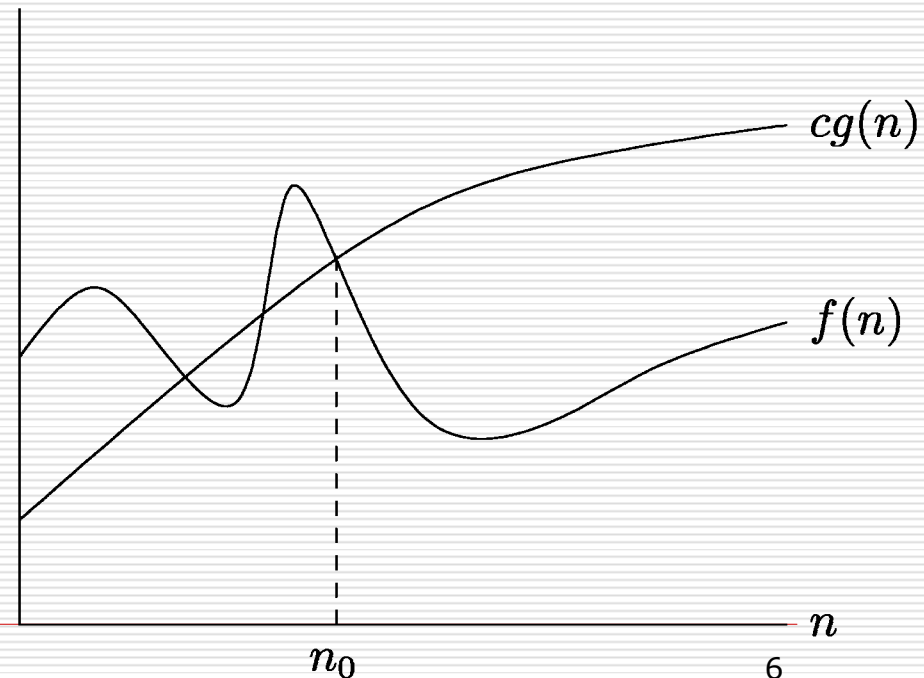
$f(n) \in O(g(n))$ ή $f(n) = O(g(n))$ αν \exists σταθερές $c, n_0 > 0$:

$$\forall n \geq n_0, f(n) \leq c g(n)$$

- $O(g(n))$ **σύνολο** συναρτήσεων με τάξη μεγέθους που **δεν υπερβαίνει** τάξη μεγέθους $g(n)$.

$$100n^3 + 10^{-5}n^3 \log n = O(n^3 \log n) = O(n^4)$$

$$10^{-10}n^2 \notin O(n)$$



Ασυμπτωτικός Συμβολισμός o

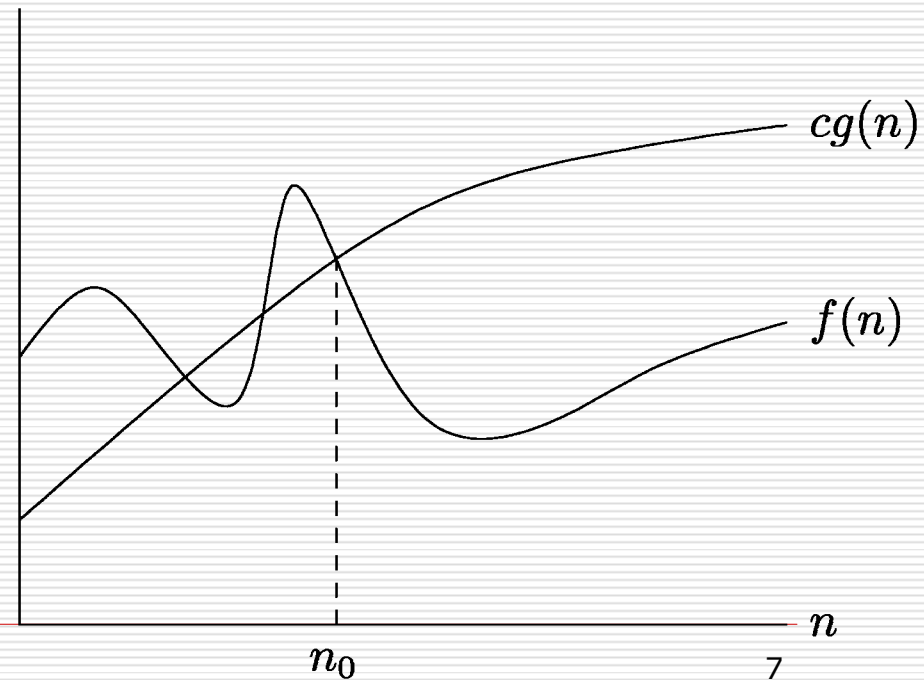
- $o(\)$ δηλώνει **άνω φράγμα** στην τάξη μεγέθους που δεν είναι ακριβές.

$$f(n) \in o(g(n)) \text{ ή } f(n) = o(g(n)) \text{ ανν } \forall c > 0, \exists n_0 > 0: \\ \forall n \geq n_0, f(n) < c g(n) \text{ ή } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

- $o(g(n))$ **σύνολο** συναρτήσεων με τάξη μεγέθους που **υπολείπεται** τάξης μεγέθους $g(n)$.

$$5n^3 \log n = o(n^4)$$

$$10n^2 \notin o(n^2).$$



Ασυμπτωτικός Συμβολισμός Ω

□ $\Omega(\)$ δηλώνει **κάτω φράγμα** στην τάξη μεγέθους.

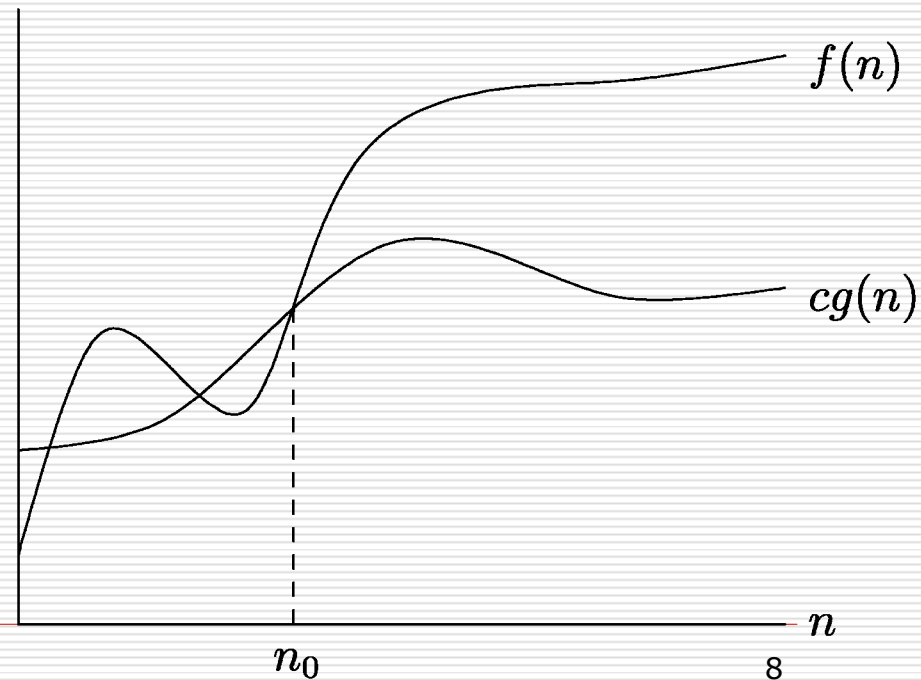
$f(n) \in \Omega(g(n))$ ή $f(n) = \Omega(g(n))$ αν \exists σταθερές $c, n_0 > 0$:

$$\forall n \geq n_0, f(n) \geq c g(n)$$

- $\Omega(g(n))$ **σύνολο** συναρτήσεων με τάξη μεγέθους που **δεν υπολείπεται** τάξης μεγέθους $g(n)$.

$$10^{-5}n^3 \log n = \Omega(n^3 \log n) = \Omega(n^3)$$

$$10^{10}n \notin \Omega(n^2).$$



Ασυμπτωτικός Συμβολισμός ω

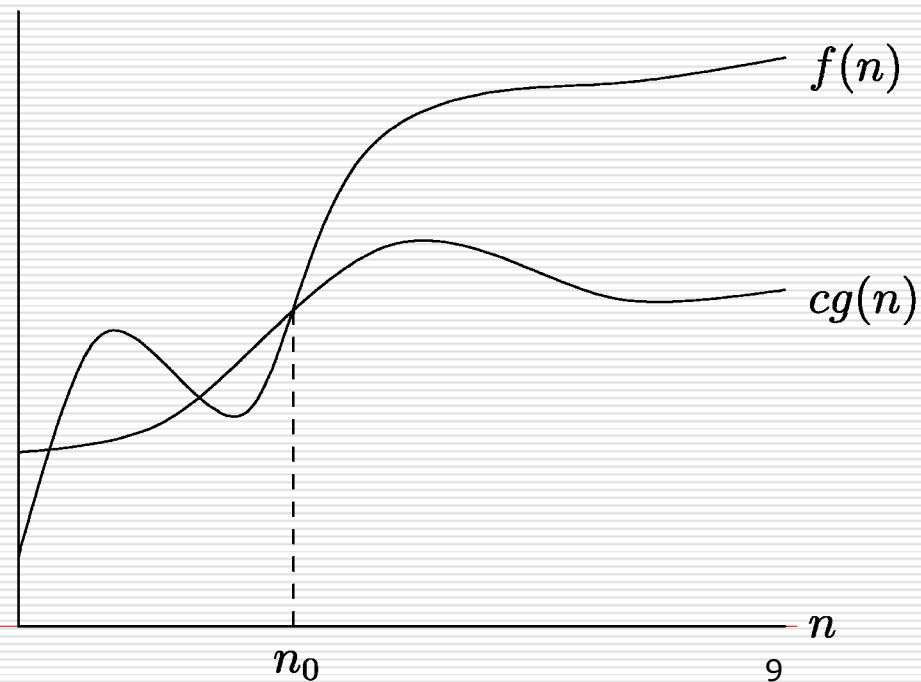
- $\omega(\)$ δηλώνει **κάτω φράγμα** στην τάξη μεγέθους που δεν είναι ακριβές.

$$f(n) \in \omega(g(n)) \text{ ή } f(n) = \omega(g(n)) \text{ ανν } \forall c > 0, \exists n_0 > 0:$$
$$\forall n \geq n_0, f(n) > c g(n) \text{ ή } \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$$

- $\omega(g(n))$ **σύνολο** συναρτήσεων με τάξη μεγέθους που **υπερβαίνει** τάξης μεγέθους $g(n)$.

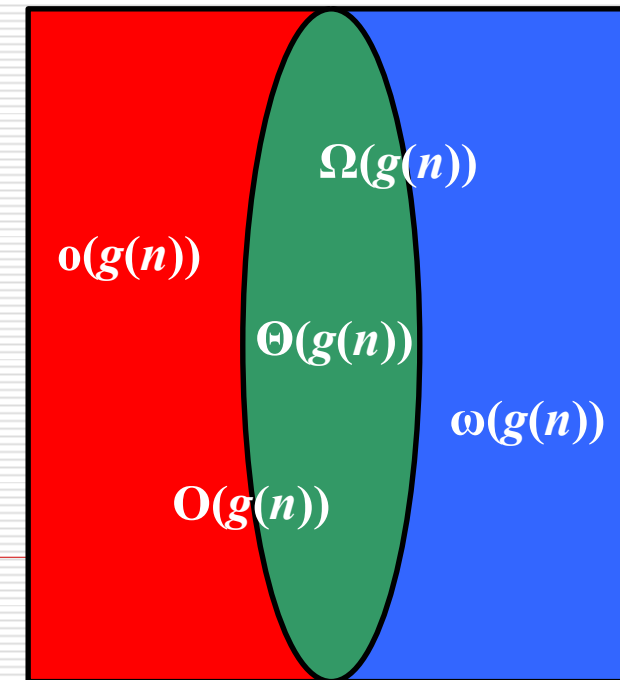
$$5n^3 \log n = \omega(n^3)$$

$$10n^2 \neq \omega(n^2).$$



Ασυμπτωτικός Συμβολισμός

- $f(n) = \Theta(g(n)) \sim$ ασυμπτωτικά $f(n) = g(n)$
 - $f(n) = O(g(n)) \sim$ ασυμπτωτικά $f(n) \leq g(n)$
 - $f(n) = o(g(n)) \sim$ ασυμπτωτικά $f(n) < g(n)$
 - $f(n) = \Omega(g(n)) \sim$ ασυμπτωτικά $f(n) \geq g(n)$
 - $f(n) = \omega(g(n)) \sim$ ασυμπτωτικά $f(n) > g(n)$
-
- Κάποιες απλές σχέσεις:
 - $O(g(n)) = o(g(n)) \cup \Theta(g(n))$
 - $\Omega(g(n)) = \omega(g(n)) \cup \Theta(g(n))$
 - $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$
 - $o(g(n)) \cap \omega(g(n)) = \emptyset$



Ασυμπτωτικός Συμβολισμός

□ Πολυώνυμο βαθμού d : $a_d n^d + a_{d-1} n^{d-1} + a_1 n + a_0 = \Theta(n^d)$

□ Κάποια αθροίσματα:

$$\sum_{i=1}^n i = \Theta(n^2), \quad \sum_{i=1}^n i^2 = \Theta(n^3), \dots, \quad \sum_{i=1}^n i^k = \Theta(n^{k+1})$$

$$\sum_{i=1}^n 1/i = \Theta(\log n), \quad \sum_{i=1}^n 2^i = \Theta(2^n)$$

□ Επίσης: $\log(n!) = \Theta(n \log n)$

□ Ιεράρχηση: $O(1) \subset O(\log^* n) \subset O(\log n) \subset O(\text{poly}(\log n)) \subset$
 $\subset O(\sqrt{n}) \subset O(n) \subset O(n \log n) \subset O(\text{poly}(n)) \subset$
 $\subset O(n^{\log n}) \subset O(2^n) \subset O(3^n) \subset O(n!) \subset O(n^n) \subset O(A(n))$

Κάποιες Ασκήσεις

- Ποιες από τις παρακάτω προτάσεις είναι αληθείς;
- $10 f(n) + 10^{100} = O(f(n))$ Αληθής
 - $f(n) + g(n) = \Theta(\min\{f(n), g(n)\})$ Ψευδής
 - $f(n) + g(n) = \Omega(\min\{f(n), g(n)\})$ Αληθής
 - $f(n) + g(n) = O(\max\{f(n), g(n)\})$ Αληθής

Κάποιες Ασκήσεις

□ Να συμπληρωθεί ο πίνακας:

$f(n)$	$g(n)$	$\Theta(g(n))$	$O(g(n))$	$o(g(n))$	$\Omega(g(n))$	$\omega(g(n))$
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		■	■		

Κάποιες Ασκήσεις

- Να βάλετε συναρτήσεις σε **αύξουσα σειρά** τάξης μεγέθους:

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

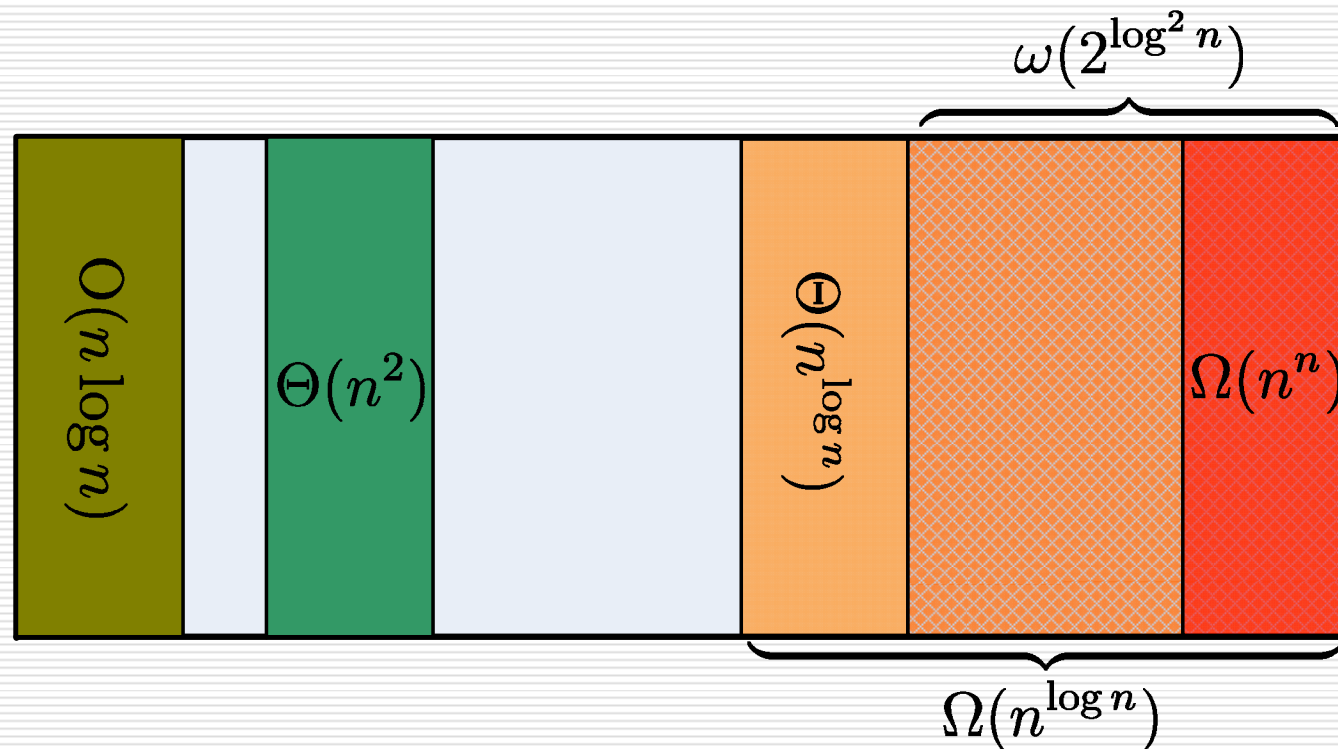
- Απάντηση:

$$\begin{aligned} n^{1/\log n} &= \Theta(1), \quad \log \log n, \quad \log^4 n, \quad (\log n)^{100} \log \log n, \\ & n^{0.1} \log \log n, \quad n^{0.6}, \\ n \log \log n, \quad \log(n!) &= \Theta(n \log n), \quad \frac{n}{\log_n 2} + n = \Theta(n \log n), \\ (\log n)^{\log n} &= \Theta(n^{\log \log n}), \quad n^{\log n}, \quad 2^{\log^3 n} = \Theta(n^{\log^2 n}), \\ 2^n, \quad 2^n + n^{2^{100}} &= \Theta(2^n), \quad 2^{5n} \end{aligned}$$

Κάποιες Ασκήσεις

- Σχεδιάστε διάγραμμα Venn για τις κλάσεις συναρτήσεων:

$$\Omega(n^n), \Theta(n^{\log n}), \omega(2^{\log^2 n}), \Omega(n^{\log n}), \Theta(n^2), O(n \log n)$$



Πρακτικά Αποδοτικοί Αλγόριθμοι

- ... έχουν **πολυωνυμική** (χρονική) πολυπλοκότητα.
 - Π.χ. $\log n$, n , $n \log n$, n^2 , n^3 , ...
 - Χρόνοι n^d , όπου **d μεγάλο**, σπάνιοι και βελτιώνονται.
- **Εκθετική** πολυπλοκότητα **απαγορευτική** για μεγάλα στιγμιότυπα! Π.χ. $100n^2 < 2^{n/5}$ για κάθε $n \geq 100$

Αύξηση **μεγεθών που λύνουμε** σε συγκεκριμένο χρόνο όταν **10πλασιάζεται** η ταχύτητα υπολογιστή:

Πολυπλ.	n	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	$1.25 (n' = n + \log 10)$

Αποδοτική Επίλυση: Κλάση P

- Αλγόριθμος πολυωνυμικού χρόνου λύνει κάθε στιγμιότυπο σε χρόνο $O(n^d)$, d σταθερά.
- **Κλάση P** : προβλήματα απόφασης που επιλύονται από αλγόριθμους πολυωνυμικού χρόνου.
 - Shortest paths, MST, max flow, min cut, min-cost flow, maximum matching, linear programming, ...
- **Αξίωμα Cook-Karp** : κλάση ευεπίλυτων προβλημάτων ταυτίζεται με κλάση P.

Αποδοτική Επίλυση: Κλάση P

- **Κλάση P** : προβλήματα που λύνονται σε **πολυωνυμικό χρόνο**.
 - Shortest paths, MST, max flow, min cut, min-cost flow, maximum matching, linear programming, ...
- **Θέση Cook-Karp** : **P** ταυτίζεται με **ευεπίλυτα προβλήματα**.
- Υπέρ θέσης Cook-Karp:
 - Συνήθως **πολυώνυμα μικρού βαθμού** (π.χ. n, n^2, n^3).
 - Διπλασιασμός υπολογιστικής ισχύος: σημαντική αύξηση στο μέγεθος στιγμιότυπων που επιλύουμε.
- Κριτική στη θέση Cook-Karp:
 - Ακραίες περιπτώσεις: θεωρείται πρακτικό το n^{100} αλλά όχι το $2^{n/100}$!
 - Γραμμικός Προγραμματισμός: **Simplex vs. Ellipsoid**.