Ασυμπτωτικός Συμβολισμός

Δημήτρης Φωτάκης

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο

Υπολογιστική Πολυπλοκότητα

- Υπολογιστική πολυπλοκότητα αλγόριθμου Α:
  - Ποσότητα υπολογιστικών πόρων που απαιτεί ο Α 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

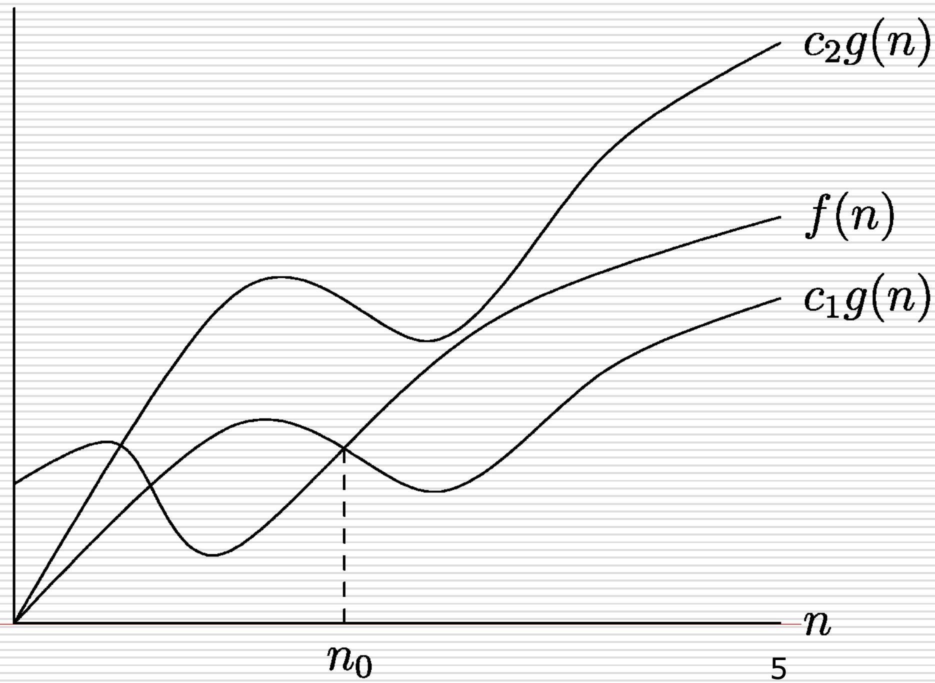
- Χρόνος εκτέλεσης αλγόριθμου Α:
  - Αύξουσα συνάρτηση του $T(n)$ που εκφράζει σε πόσο χρόνο ολοκληρώνεται ο Α όταν εφαρμόζεται σε στιγμ. μεγέθους n.
- Ενδιαφέρει η τάξη μεγέθους $T(n)$ και όχι ακριβής εκτίμηση $T(n)$.
  - Ακριβής εκτίμηση είναι συχνά δύσκολη και εξαρτάται από υπολογιστικό περιβάλλον, 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.
- Θ( ) δηλώνει την **ακριβή εκτίμηση** τάξης μεγέθους.

$$f(n) \in \Theta(g(n)) \text{ ή } f(n) = \Theta(g(n)) \text{ ανν } \exists \text{σταθερές } c_1, c_2, n_0 > 0 :$$
$$\forall n \geq n_0 \,, \quad c_1\, g(n) \leq f(n) \leq c_2\, g(n)$$

  - $\Theta(g(n))$ 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 Ο

- Ο( ) δηλώνει **άνω φράγμα** στην τάξη μεγέθους.

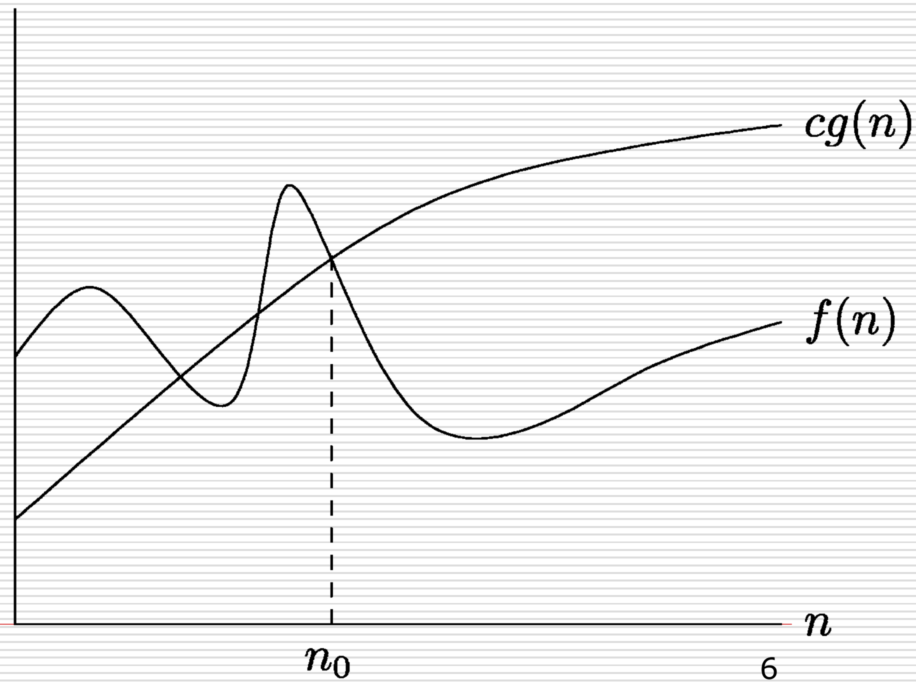
  $f(n) \in \mathrm{O}(g(n))$ ή $f(n) = \mathrm{O}(g(n))$ ανν $\exists$σταθερές $c, n_0 > 0:$

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

  - Ο(g(n)) σύνολο συναρτήσεων με τάξη μεγέθους που δεν υπερβαίνει τάξη μεγέθους g(n).

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

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



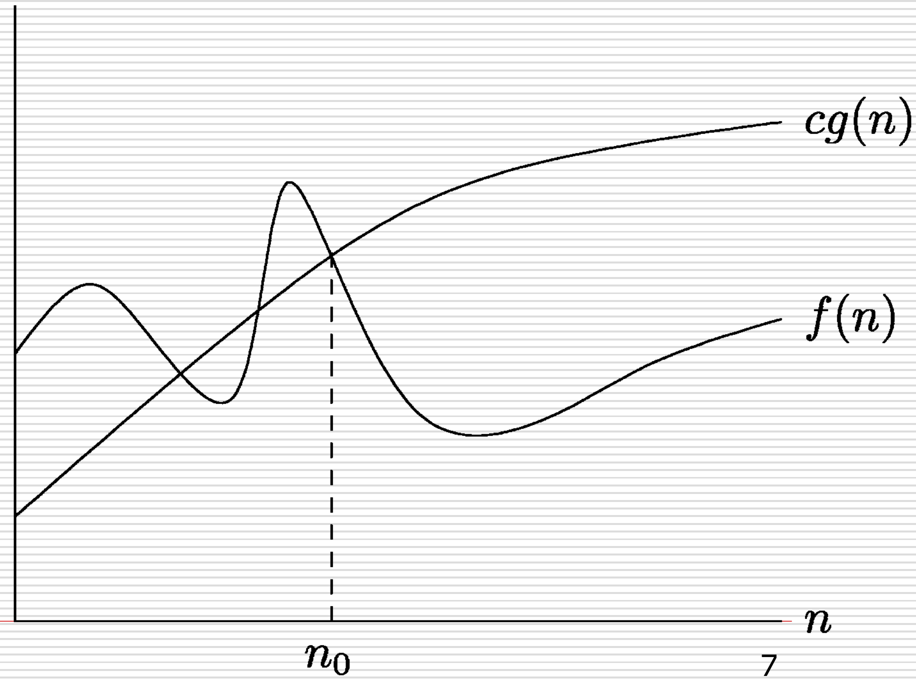
Ασυμπτωτικός Συμβολισμός o

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

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

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

    $5n^3 \log n = \mathrm{o}(n^4)$
    $10n^2 \notin \mathrm{o}(n^2)$.



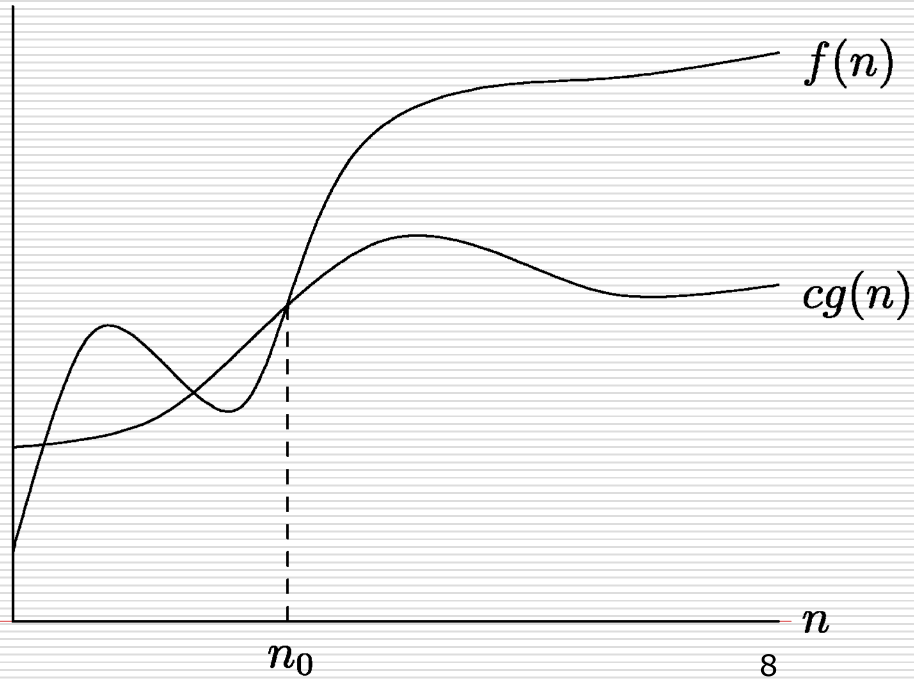
Ασυμπτωτικός Συμβολισμός Ω

- Ω( ) δηλώνει **κάτω φράγμα** στην τάξη μεγέθους.

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

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

  - $\Omega(g(n))$ 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

- ω( ) δηλώνει **κάτω φράγμα** στην τάξη μεγέθους που δεν είναι ακριβές.

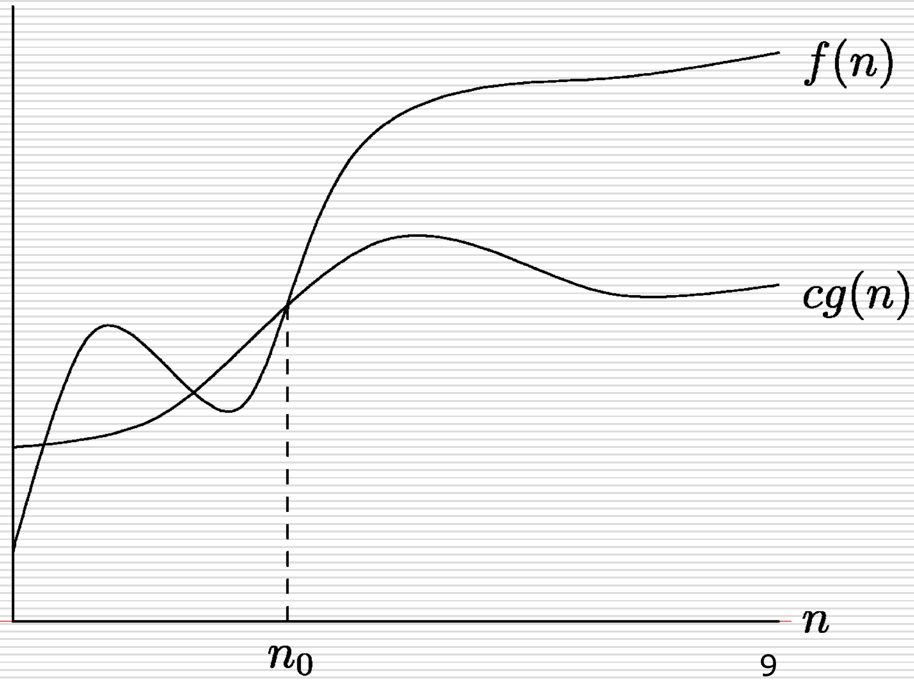
  $f(n) \in \omega(g(n))$ ή $f(n) = \omega(g(n))$ ανν $\forall c > 0, \exists n_0 > 0:$

  $$\forall n \geq n_0 \,, \quad f(n) > c\,g(n) \quad \text{ή} \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

  - ω(*g*(*n*)) σύνολο συναρτήσεων με τάξη μεγέθους που υπερβαίνει τάξης μεγέθους *g*(*n*).

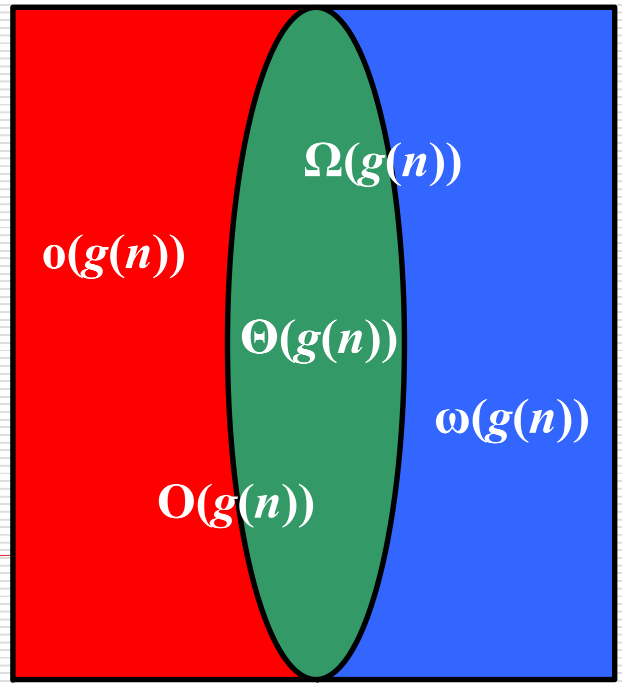
    $5n^3 \log n = \omega(n^3)$

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



Ασυμπτωτικός Συμβολισμός

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



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:

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

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

- Επίσης: $\log(n!) = \Theta(n \log n)$
- Ιεράρχηση: $\mathrm{O}(1) \subset \mathrm{O}(\log^* n) \subset \mathrm{O}(\log n) \subset \mathrm{O}(\mathrm{poly}(\log n)) \subset$
  $\subset \mathrm{O}(\sqrt{n}) \subset \mathrm{O}(n) \subset \mathrm{O}(n \log n) \subset \mathrm{O}(\mathrm{poly}(n)) \subset$
  $\subset \mathrm{O}(n^{\log n}) \subset \mathrm{O}(2^n) \subset \mathrm{O}(3^n) \subset \mathrm{O}(n!) \subset \mathrm{O}(n^n) \subset \mathrm{O}(A(n))$

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:

$f(n)$	$g(n)$	$\Theta(g(n))$	$\mathrm{O}(g(n))$	$o(g(n))$	$\Omega(g(n))$	$\omega(g(n))$
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:

$$
n^{1/\log n} = \Theta(1),\ \ \log\log n,\ \ \log^4 n,\ \ (\log n)^{100} \log\log n,
$$

$$
n^{0.1} \log\log n,\ \ n^{0.6},
$$

$$
n \log\log n,\ \ \log(n!) = \Theta(n \log n),\ \ \frac{n}{\log_n 2} + n = \Theta(n \log n),
$$

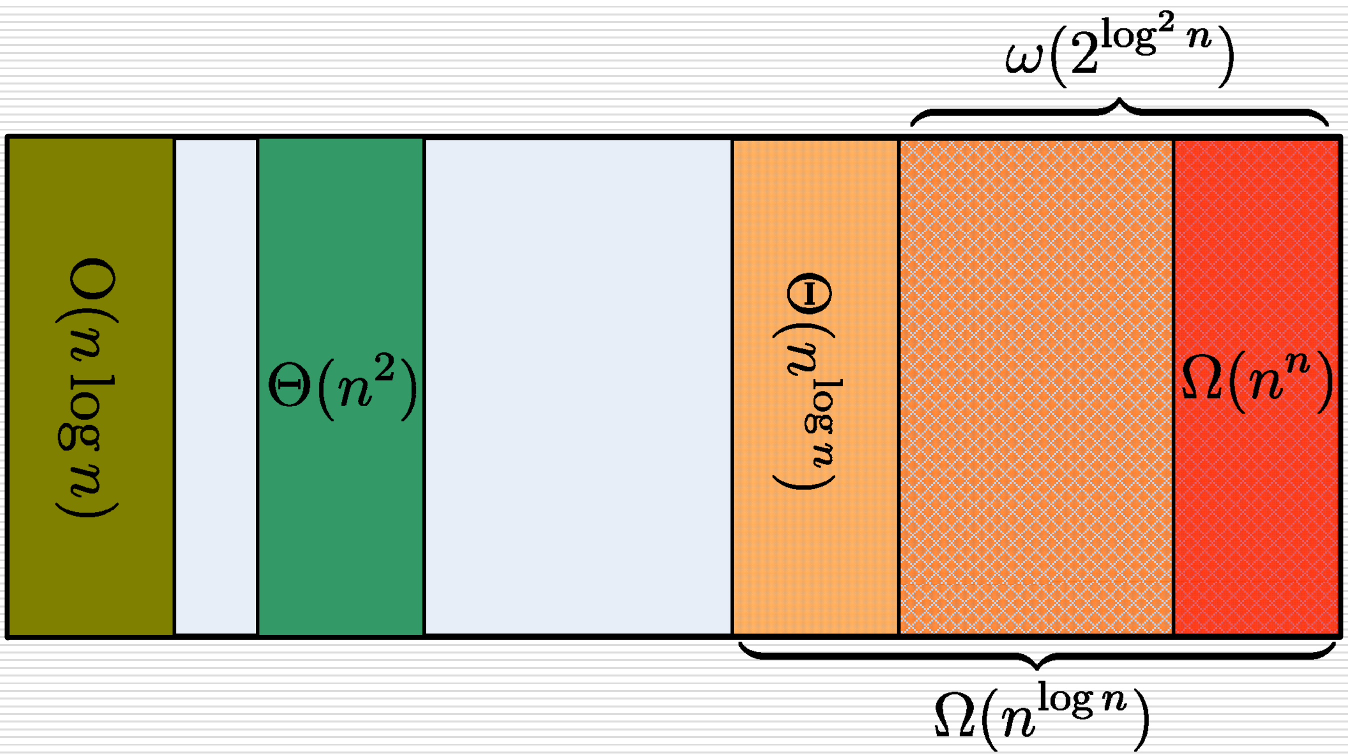
$$
(\log n)^{\log n} = \Theta(n^{\log\log n}),\ \ n^{\log n},\ \ 2^{\log^3 n} = \Theta(n^{\log^2 n}),
$$

$$
2^n,\ \ 2^n + n^{2^{100}} = \Theta(2^n),\ \ 2^{5n}
$$

Κάποιες Ασκήσεις

- Σχεδιάσετε 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, …
- **Θέση 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.