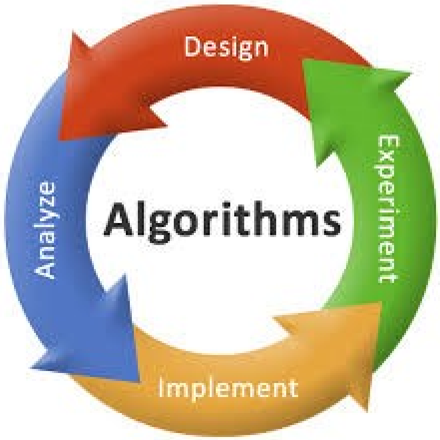
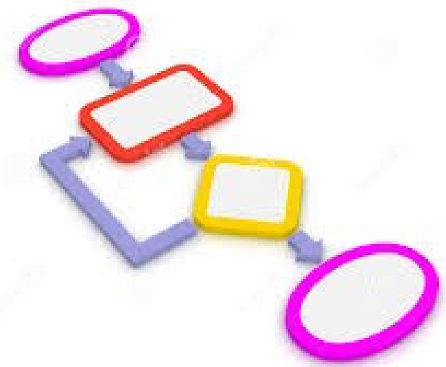




ΑΛΓΟΡΙΘΜΟΙ

Δρ. Χάρης Κουζινόπουλος

Εθνικό Κέντρο Έρευνας και Τεχνολογικής Ανάπτυξης

Πανεπιστήμιο Μακεδονίας

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ
Τμήμα Εφαρμοσμένης Πληροφορικής

CMOR
Computational Methodologies & Operations Research

Αλγόριθμοι

- Ο όρος αλγόριθμος είναι παραφθορά του ονόματος al-Khwarizmi (Αλ Χουαρίζμι), ο οποίος ήταν μαθηματικός του ένατου αιώνα, του οποίου το βιβλίο για τους ινδουιστικούς αριθμούς είναι η βάση της σύγχρονης δεκαδικής γραφής
- Αρχικά η λέξη αλγορισμός χρησιμοποιήθηκε για να χαρακτηρίσει τους κανόνες για την εκτέλεση αριθμητικής με χρήση δεκαδικών αριθμών.
- Ο αλγορισμός εξελίχθηκε στη λέξη αλγόριθμος τον δέκατο όγδοο αιώνα

Παράδειγμα υπολογιστικού προβλήματος: Εύρεση μέγιστου στοιχείου πεπερασμένης ακολουθίας

- Πως μπορεί να λυθεί;

Παράδειγμα υπολογιστικού προβλήματος: Εύρεση μέγιστου στοιχείου πεπερασμένης ακολουθίας

- Ορισμός ως προσωρινό μέγιστο τον πρώτο ακέραιο στην ακολουθία. (Προσωρινός μέγιστος θα είναι ο μεγαλύτερος ακέραιος αριθμός που εξετάστηκε σε οποιοδήποτε στάδιο της διαδικασίας)
- Σύγκριση του επόμενου ακέραιο αριθμού της ακολουθίας με το προσωρινό μέγιστο - αν είναι μεγαλύτερος από το προσωρινό μέγιστο, ορίζουμε το προσωρινό μέγιστο ίσο με αυτόν τον ακέραιο αριθμό.
- Επανάληψη του προηγούμενου βήματος εάν υπάρχουν περισσότεροι ακέραιοι αριθμοί στην ακολουθία
- Ολοκλήρωση όταν δεν υπάρχουν ακέραιοι αριθμοί στην ακολουθία. Το προσωρινό μέγιστο σε αυτό το σημείο είναι ο μεγαλύτερος ακέραιος στην ακολουθία

Παράδειγμα υπολογιστικού προβλήματος: Εύρεση μέγιστου στοιχείου πεπερασμένης ακολουθίας

Διαδικασία max(a , a ,...,a : ακέραιοι)

- max := a
- **Για** i := 2 **έως** n

  εάν max < a **τότε** max := a
- **επιστροφή** max{max είναι το μεγαλύτερο στοιχείο}

Ορισμοί ΜΚΔ-ΕΚΠ

Μέγιστος Κοινός Διαιρέτης:

ΜΚΔ(x,y) = μέγιστος ακέραιος $k \geq 1$
τέτοιος ώστε: $k \mid x$ και $k \mid y$

Αν ΜΚΔ(x,y)=1 τότε οι αριθμοί x,y λέγονται πρώτοι μεταξύ τους.

Ελάχιστο Κοινό Πολλαπλάσιο:

ΕΚΠ(x,y) = ελάχιστος ακέραιος $k \geq 1$
τέτοιος ώστε: $x \mid k$ και $y \mid k$

Βασικές Ιδιότητες ΜΚΔ-ΕΚΠ

1. $|a \cdot b|$ = ΜΚΔ(a,b) $\cdot$ ΕΚΠ(a,b)
2. Θεώρημα του Bézout: Αν a και b θετικοί ακέραιοι τότε υπάρχουν ακέραιοι s και t τέτοιοι ώστε: ΜΚΔ(a,b) = $sa + tb$
3. Αν ΜΚΔ(a,b)=1 και $a \mid bc$, τότε $a \mid c$.
4. Αν ο p είναι πρώτος και $p \mid a_1 \times a_2 \times \ldots \times a_n$, όπου κάθε a_i είναι ακέραιος, τότε $p \mid a_i$ για κάποιο i.
5. Θεώρημα Ευκλείδη: Αν a,b θετικοί ακέραιοι και r το υπόλοιπο της διαίρεσης του a με το b, τότε:

ΜΚΔ(a,b)=ΜΚΔ(b,r)

Ο Αλγόριθμος του Ευκλείδη

- **Αν *a* και *b* θετικοί ακέραιοι τότε μπορούμε να βρούμε γρήγορα τον ΜΚΔ τους με βάση το προηγούμενο θεώρημα.**
- **Ιδέα:**
  - **Αν *r* το υπόλοιπο της διαίρεσης των *a,b* τότε έχουμε:**

    $$\text{ΜΚΔ}(a,b)=\text{ΜΚΔ}(b,r)$$

  - **Άρα αρκεί να βρούμε τον ΜΚΔ των *b,r*.**
  - **Επαναλαμβάνουμε με νέα διαίρεση μέχρι να καταλήξουμε στην τετριμμένη περίπτωση:**

    $$\text{ΜΚΔ}(x,0)=x$$

Παράδειγμα υπολογιστικού προβλήματος: Ο αλγόριθμος του Ευκλείδη

```
int euclid_gcd(int a, int b) {
    int dividend = a >= b ? a : b;
    int divisor = a <= b ? a : b;
    while(divisor != 0) {
        int remainder = dividend % divisor;
        dividend = divisor;
        divisor = remainder;
    }
    return dividend;
}
```

Παράδειγμα (1ο)

- $a = 15$, $b = 12$

a	b	q	r	$a = b \cdot q + r$
15	12	1	3	$15 = 12 \cdot 1 + 3$
12	3	4	0	$12 = 3 \cdot 4 + 0$
3	0			

- Άρα ΜΚΔ(15, 12) = ΜΚΔ(12, 3) = ΜΚΔ(3, 0) = 3

Παράδειγμα (2ο)

- a = 35731, b = 24689

a	b	q	r	$a = b \cdot q + r$
35731	24689	1	11042	$35731 = 24689 \cdot 1 + 11042$
24689	11042	2	2605	$24689 = 11042 \cdot 2 + 2605$
11042	2605	4	622	$11042 = 2605 \cdot 4 + 622$
2605	622	4	117	$2605 = 622 \cdot 4 + 117$
622	117	5	37	$622 = 117 \cdot 5 + 37$
117	37	3	6	$117 = 37 \cdot 3 + 6$
37	6	6	1	$37 = 6 \cdot 6 + 1$
6	1	6	0	$6 = 1 \cdot 6 + 0$
1	0			

Άρα ΜΚΔ(35731, 24689) = … = ΜΚΔ(1, 0) = 1 δηλαδή οι αριθμοί αυτοί είναι πρώτοι μεταξύ τους.

Τι είναι η χρονική πολυπλοκότητα

- Έστω ότι έχω δώσει ένα στυλό σε κάποιον στο αμφιθέατρο αλλά δε θυμάμαι σε ποιον
- $O(n^2)$: Ρωτάω το πρώτο άτομο στην τάξη αν έχει το στυλό μου. Επίσης, ρωτάω αυτό το άτομο αν τα υπόλοιπα n-1 άτομα στην τάξη έχουν αυτό το στυλό
- O(n): Ρωτάω τον κάθε φοιτητή ξεχωριστά
- O(logn): Τώρα χωρίζω την τάξη σε δύο ομάδες και μετά ρωτάω: «Είναι το στυλό στην αριστερή πλευρά ή στη δεξιά πλευρά της τάξης;» Μετά παίρνω αυτή την ομάδα και τη χωρίζω στα δύο και ξαναρωτάω, και ούτω καθεξής. Επαναλαμβάνω τη διαδικασία μέχρι να μείνω με έναν μαθητή που έχει το στυλό μου

Θεωρητική ανάλυση χρονικής πολυπλοκότητας

- Η χρονική αποδοτικότητα αναλύεται προσδιορίζοντας τον αριθμό των επαναλήψεων της **βασικής πράξης** ως συνάρτηση του **μεγέθους εισόδου**
- *Βασική πράξη:* η πράξη που συνεισφέρει περισσότερο από τις άλλες στο χρόνο εκτέλεσης του αλγορίθμου

$$T(n) = c_{op}c(n)$$

Χρόνος εκτέλεσης για ένα μέγεθος εισόδου = χρόνος εκτέλεσης βασικής πράξης * πλήθος επαναλήψεων βασικής πράξης

Μέγεθος εισόδου και βασική πράξη

Πρόβλημα	***Μέγεθος εισόδου***	***Βασική πράξη***
Αναζήτηση κλειδιού σε λίστα με *n* αντικείμενα	Το πλήθος *n των αντικειμένων*	Συγκρίσεις κλειδιών
Πολλαπλασιασμός πινάκων με πραγματικούς αριθμούς	Διαστάσεις των πινάκων	Πολλαπλασιασμός πραγματικών αριθμών
Προβλήματα με γράφους	Πλήθος κορυφών ή/και ακμών	Η επίσκεψη ενός κόμβου ή η διάσχιση μίας ακμής

Καλύτερη, μέση, χειρότερη περίπτωση

Σε κάποιους αλγορίθμους η αποδοτικότητα εξαρτάται από τον τύπο της εισόδου:

- Χειρότερη περίπτωση: *W(n)* – το μέγιστο από όλες τις εισόδους μεγέθους *n*
- Καλύτερη περίπτωση: *B(n)* – το ελάχιστο από όλες τις εισόδους μεγέθους *n*
- Μέση περίπτωση: *A(n)* – ο μέσος όρος από όλες τις εισόδους μεγέθους *n*
    - Το πλήθος εκτελέσεων της βασικής πράξης σε μία τυπική είσοδο

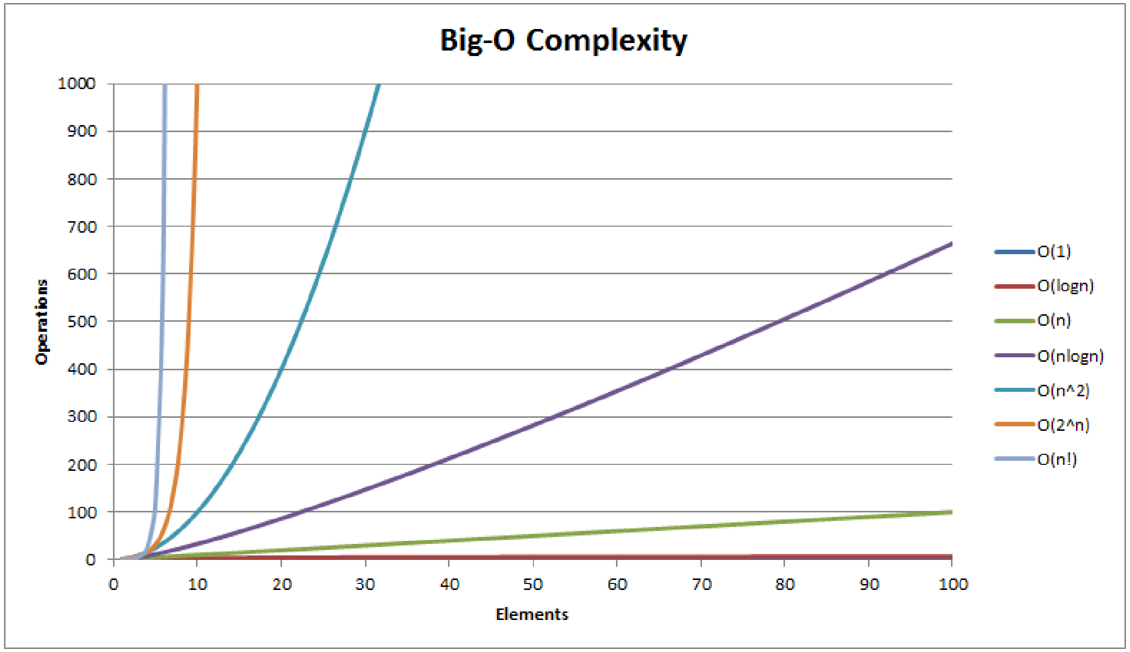
Καλύτερη, μέση, χειρότερη περίπτωση

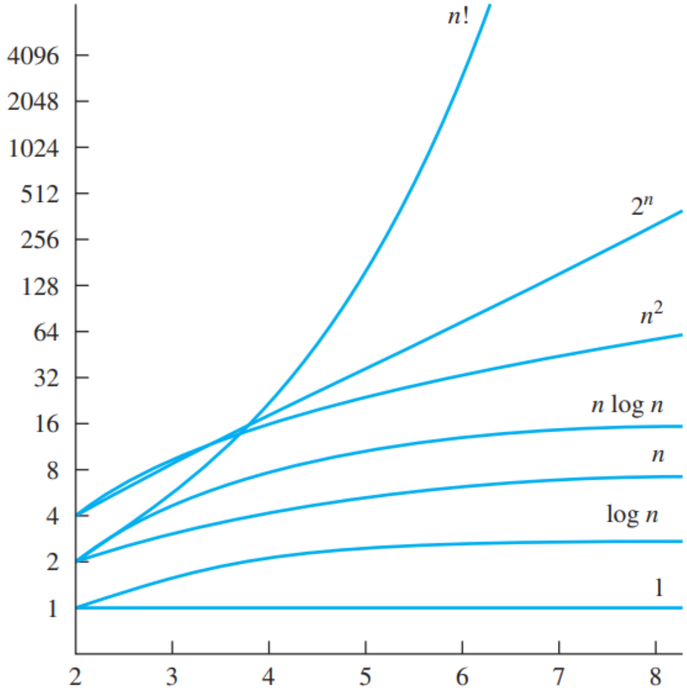
Algorithms	Best Case			Average Case			Worst Case		
	Small Data Set()	Average Data Set	Large Data Set	Small Data Set()	Average Data Set	Large Data Set	Small Data Set()	Average Data Set	Large Data Set
Bubble Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
E BS	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
Enhanced BS	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
Enhanced SS									
Insertion Sort	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
Quick Sort	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$			
Enhanced QS	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Merge Sort	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$

Τάξη μεγέθους

n	$\log_2 n$	n	$n\log_2 n$	n^2	n^3	2^n	$n!$
10	3.3	10	$3.3 * 10$	10^2	10^3	10^3	
10^2	6.6	10^2	$6.6 * 10^2$	10^4	10^6	$1.3 * 10^{30}$	$3.6 * 10^6$
10^3	10	10^3	10^4	10^6	10^9		$9.3 * 10^{157}$
10^4	13	10^4	$13 * 10^4$	10^8	10^{12}		
10^5	17	10^5	$17 * 10^5$	10^{10}	10^{15}		
10^6	20	10^6	$20 * 10^6$	10^{12}	10^{18}		

Τάξη μεγέθους





Ασυμπτωτικός ρυθμός αύξησης

Παράδειγμα:

Αλγόριθμος	**Πολυπλοκότητα**
A1	1000n
A2	200nlogn
A3	10n
A4	2

Ποιος αλγόριθμος είναι ταχύτερος/αποδοτικότερος;

Ασυμπτωτικός ρυθμός αύξησης

Παράδειγμα:

Αλγόριθμος	**Πολυπλοκότητα**
A1	1000n
A2	200nlogn
A3	10n
A4	2^n

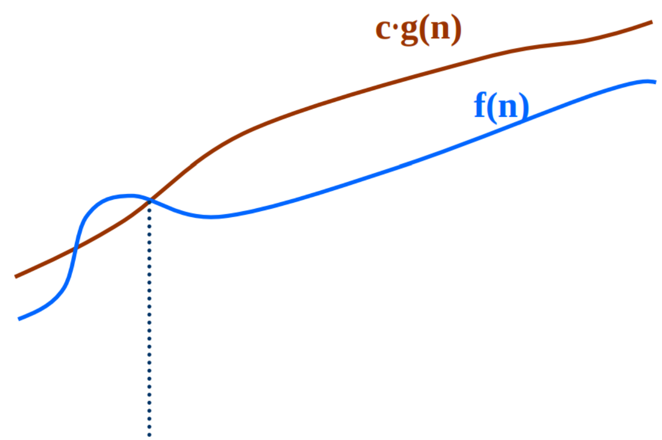
Ποιος αλγόριθμος είναι ταχύτερος/αποδοτικότερος; Εξαρτάται!

n	**Αποδοτικότερος**
0 < n < 10	A4
10 <= n <= 100	A3
n > 100	A1

Ασυμπτωτικός ρυθμός αύξησης

Τρόπος σύγκρισης συναρτήσεων που αγνοεί τους σταθερούς παράγοντες και τα μικρά μεγέθη εισόδου – περιγράφει ένα ασυμπτωτικό ανώτατο όριο

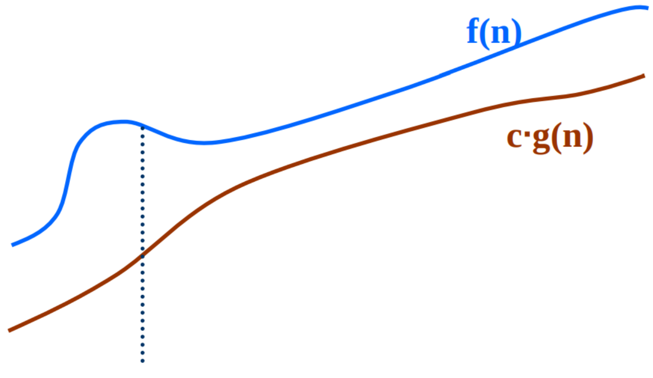
•*O(g(n))*: η κλάση των συναρτήσεων *f(n)* που μεγαλώνουν *όχι γρηγορότερα* από μια συνάρτηση *g(n)*



Ασυμπτωτικός ρυθμός αύξησης

Τρόπος σύγκρισης συναρτήσεων που αγνοεί τους σταθερούς παράγοντες και τα μικρά μεγέθη εισόδου – περιγράφει ένα ασυμπτωτικό κάτω όριο

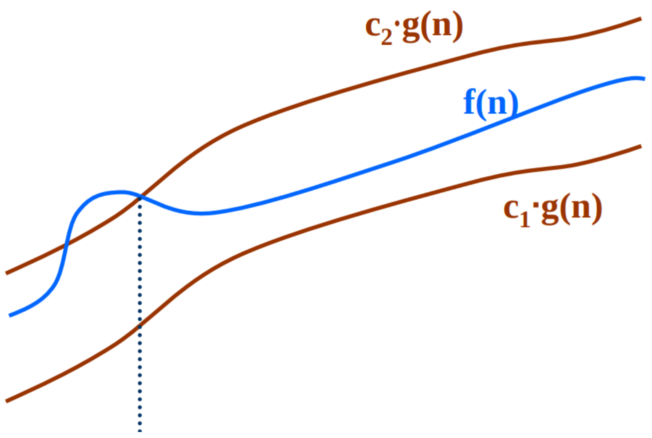
• *Ω(g(n))*: η κλάση των συναρτήσεων *f(n)* που μεγαλώνουν *τουλάχιστον τόσο γρήγορα όσο* μια συνάρτηση *g(n)*



Ασυμπτωτικός ρυθμός αύξησης

Τρόπος σύγκρισης συναρτήσεων που αγνοεί τους σταθερούς παράγοντες και τα μικρά μεγέθη εισόδου

- *Θ(g(n))*: η κλάση των συναρτήσεων *f(n)* που μεγαλώνουν *με τον ίδιο ρυθμό* όπως μια συνάρτηση *g(n)*



Ασυμπτωτικός ρυθμός αύξησης

Άσκηση 1:
Υποθέστε ότι έχετε δύο αλγορίθμους A_1 και A_2 με αριθμό στοιχειωδών λειτουργιών $T(A_1) = 80n$ και $T(A_2) = 4n^3$, για την επίλυση ενός προβλήματος μεγέθους n. Υποθέστε επίσης ότι έχετε έναν υπολογιστή που εκτελεί 2^4 στοιχειώδεις λειτουργίες ανά δευτερόλεπτο

α) Ποιος είναι ο πραγματικός χρόνος εκτέλεσης των A_1 και A_2 για $n=2^5$;

β) Για κάθε έναν από τους A_1 και A_2 ποια είναι η μέγιστη τιμή του n για την οποία ο υπολογιστής θα δώσει αποτέλεσμα σε 3 λεπτά υπολογισμού;

Ασυμπτωτικός ρυθμός αύξησης

Άσκηση 1:

α)

t_{A1} = (80 * 2^5 λειτουργίες) / (2^4 λειτουργίες / sec) = 160 sec

t_{A2} = (4 * $(2^5)^3$ λειτουργίες) / (2^4 λειτουργίες / sec) = 8192 sec

β)

(80n λειτουργίες) / (2^4 λειτουργίες / sec) = 3 * 60 secs ⇔ n = 36

($4n^3$ λειτουργίες) / (2^4 λειτουργίες / sec) = 3 * 60 secs ⇔n = 8