Αλγόριθμοι και Πολυπλοκότητα

Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
Πανεπιστήμιο Αιγαίου

Αλγόριθμοι και Πολυπλοκότητα

- 4 ώρες **θεωρία** (Δ. Φωτάκης, **Τετάρτη 17-19** και **Πέμπτη 13-15**):
  - **Γραπτή Πρόοδος**: αρχές Ιουνίου, 20% βαθμού (ΒΠ);;;
  - **Τελική εξέταση**: τέλος Ιουνίου, 50% βαθμού (ΒΤΕ), **ΒΤΕ** ≥ 4
- **Εργαστήριο** (Ελ. Κωνσταντίνου, **Τρίτη 12-14** και **Τρίτη 14-16**):
  - **6 Ασκήσεις**: Εκφώνηση **1 εβδομάδα πριν**, προετοιμασία, υλοποίηση, **εξέταση, και παράδοση** στο εργαστήριο (με καθοδήγηση).
  - **Υποχρεωτικό**: 30% Βαθμού (ΒΕργ), **ΒΕργ** ≥ 5
  - **Καμία παράταση** στην παράδοση των ασκήσεων!
  - Διαλέξεις 25/4 και 26/4 (εργ.): πως αποθηκεύουμε και προσπελαύνουμε γραφήματα.
  - **1η Άσκηση**: ανακοίνωση 27/4, παράδοση 8/5.
- **Τελικός Βαθμός**: $0.5 \times$ ΒΤΕ $+ 0.2 \times$ ΒΠ $+ 0.3 \times$ ΒΕργ εφόσον ΒΕργ ≥ 5 **και** ΒΤΕ ≥ 4.

Βιβλιογραφία

- Δ. Φωτάκης, Αλγόριθμοι και Πολυπλοκότητα, 2005.
- Π. Μποζάνης, Αλγόριθμοι: Σχεδιασμός και Ανάλυση, Εκδ. Τζιόλα, 2005.
- Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms, 2nd Edition, MIT Press, 2002.
- Brassard and Bratley, Algorithmics: Theory and Practice, Prentice-Hall, 1998.
- Garey and Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, 1979.
- **Οτιδήποτε** άλλο (ασκήσεις, παραδείγματα, εφαρμογές) αναφέρεται στις **διαλέξεις**.
- Σημειώσεις, ανακοινώσεις, άλλο υλικό: http://www.icsd.aegean.gr/lecturers/fotakis/algorithms.html
- **Παρακολούθηση** διαλέξεων και ενεργή **συμμετοχή** στο εργαστήριο.

Αντικείμενο - Ύλη

- Βασικές **τεχνικές** σχεδιασμού και ανάλυσης αλγορίθμων:
  - Διαίρει-και-Βασίλευε
  - Δυναμικός Προγραμματισμός.
  - Απληστία.
- Αλγόριθμοι **γραφημάτων**:
  - Αναζήτηση Πρώτα σε Πλάτος και Αναζήτηση Πρώτα σε Βάθος.
  - Ελάχιστα Επικαλύπτοντα Δέντρα.
  - Συντομότερα Μονοπάτια.
- Εισαγωγή στην **Υπολογιστική Πολυπλοκότητα**:
  - Μηχανές Turing και υπολογισιμότητα.
  - Κλάσεις πολυπλοκότητας, αναγωγές και πληρότητα.
  - Οι κλάσεις **P** και **NP**, **NP**-πληρότητα.

Αλγόριθμος - Πρόβλημα - Στιγμιότυπο

- **Αλγόριθμος**: "Συνταγή" για την επίλυση ενός **προβλήματος**.
  Σαφώς ορισμένη διαδικασία για την **επίλυση προβλήματος** σε **πεπερασμένο** χρόνο από υπολογιστική **μηχανή**.
  Ευκλείδης ΜΚΔ, αριθμητικές πράξεις, ταξινόμηση, αναζήτηση.
- **Πρόβλημα**: Μετασχηματισμός **εισόδου** (input) σε **έξοδο** (output).
  Ορίζει **ακριβώς** μορφή δεδομένων εισόδου και εξόδου.
  Πολλαπλασιασμός, συντομότερο μονοπάτι, μέγιστη ροή.
- **Στιγμιότυπο**: Δεδομένα που συμφωνούν με περιορισμούς προβλήματος.
  Έγκυρη είσοδος προβλήματος.
  5×20, γράφημα με μήκη, γράφημα με χωρητικότητες.
- **Ορθότητα** αλγορίθμου: Λύνει / απαντάει σωστά σε όλα τα στιγμιότυπα.
  Λάθος: **αντι-παράδειγμα**. Ορθότητα: **μαθηματική απόδειξη**.
- Προβλήματα λύνονται από **πολλούς σωστούς** αλγόριθμους:
  Ποιός είναι ο **καλύτερος** (για συγκεκριμένη εφαρμογή);

Αλγόριθμος - Πρόβλημα - Στιγμιότυπο

- **(Υπολογιστικό) Πρόβλημα** συνίσταται σε άπειρο σύνολο **στιγμιοτύπων**.
- **Στιγμιότυπο** είναι η **είσοδος** (μαθηματικό αντικείμενο) για την οποία ρωτάμε **ερώτηση** και περιμένουμε **απάντηση**.
- Δύο είδη προβλημάτων:
  - **Απόφασης**: απαντήσεις **ΝΑΙ** ή **ΟΧΙ**.
  - **Βελτιστοποίησης**: καλύτερη **αποδεκτή λύση**.
- **Αλγόριθμος** απαντάει πάντα **σωστά** σε ερώτηση.

Προβλήμα Βελτιστοποίησης

- Πρόβλημα **βελτιστοποίησης** Π:
  - Σύνολο **στιγμιοτύπων** Σ_Π.
  - $\forall \sigma \in \Sigma_\Pi$, σύνολο **αποδεκτών** (εφικτών) **λύσεων** $\Lambda_\Pi(\sigma)$.
  - $\forall \sigma \in \Sigma_\Pi$, **αντικειμενική συνάρτηση** $f_\sigma : \Lambda_\Pi(\sigma) \mapsto \mathbb{R}$.
- Δεδομένου $\sigma \in \Sigma_\Pi$, ζητείται $\lambda^*_\sigma \in \Lambda_\Pi(\sigma)$:
  $\forall\, \lambda \in \Lambda_\Pi(\sigma), \quad f_\sigma(\lambda^*_\sigma) \geq f_\sigma(\lambda)$ πρόβλημα **μεγιστοποίησης**
  $\forall\, \lambda \in \Lambda_\Pi(\sigma), \quad f_\sigma(\lambda^*_\sigma) \leq f_\sigma(\lambda)$ πρόβλημα **ελαχιστοποίησης**
- λ^*_σ **βέλτιστη λύση** και $f_\sigma(\lambda^*_\sigma)$ **βέλτιστη αντικειμενική τιμή**.
- Πρόβλημα **Συνδυαστικής** Βελτιστοποίησης: υπάρχει **πεπερασμένο σύνολο** αποδεκτών λύσεων που εγγυημένα περιλαμβάνει βέλτιστη λύση.
  Π.χ. διακριτός χώρος λύσεων.

Παραδείγματα Προβλημάτων

- **Πρόβλημα Προσπελασιμότητας**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα $G(V, E)$ και διακεκριμένες κορυφές s και t.
  - **Ερώτηση**: Υπάρχει μονοπάτι από s στο t;
- **Πρόβλημα Συντομότερου Μονοπατιού**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα με μήκη στις ακμές $G(V, E, w)$ και διακεκριμένες κορυφές s και t.
  - **Ερώτηση**: Ποιο είναι το συντομότερο $s - t$ μονοπάτι;

Παραδείγματα Προβλημάτων

- **Πρόβλημα κύκλου Hamilton**:
  - **Στιγμιότυπο**: Γράφημα $G(V, E)$.
  - **Ερώτηση**: Υπάρχει κύκλος Hamilton στο G (κύκλος που διέρχεται από κάθε κορυφή ακριβώς μία φορά);
- **Πρόβλημα Περιοδεύοντος Πωλητή**:
  - **Στιγμιότυπο**: Σύνολο $= \{1, \ldots, n\}$ σημείων και αποστάσεις $d(i, j)$ μεταξύ κάθε ζεύγους διαφορετικών σημείων.
  - **Ερώτηση**: Ποια μετάθεση π του N ελαχιστοποιεί

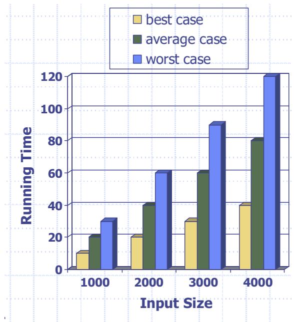
$$d(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} d(\pi(i), \pi(i+1))$$

Παραδείγματα Προβλημάτων

- **Πρόβλημα Μέγιστης Ροής**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα με χωρητικότητες στις ακμές $G(V, E, u)$ και διακεκριμένες κορυφές s και t.
  - **Ερώτηση**: Πόση είναι η μέγιστη ροή από το s στο t που δεν παραβιάζει τις χωρητικότητες (και πως δρομολογείται);
- **Πρόβλημα Ροής Ελάχιστου Κόστους**:
  - **Στιγμιότυπο**: Κατευθυνόμενο γράφημα με χωρητικότητες και κόστη στις ακμές $G(V, E, u, c)$, διακεκριμένες κορυφές s και t, απαίτηση d. Κόστος ροής f_e σε ακμή e: $f_e c_e$ (γραμμικό κόστος).
  - **Ερώτηση**: Ποιο είναι το μικρότερο κόστος για την αποστολή d μονάδων ροής από το s στο t (και ποια δρομολόγηση το επιτυγχάνει);

Ανάλυση Αλγορίθμων

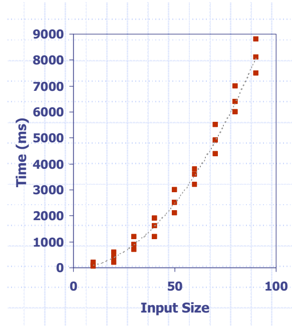
- **Πολυπλοκότητα**: ποσότητα **υπολογιστικών πόρων** που απαιτεί ο αλγόριθμος σαν **αύξουσα συνάρτηση** μεγέθους στιγμιότυπου που επιλύει.
  χρόνος, **μνήμη**, επεξεργαστές, επικοινωνία μέσω δικτύου
- **Μέγεθος στιγμιότυπου** n: Αριθμός bits για αναπαράσταση στη μνήμη.
  Πλήθος **βασικών συνιστωσών** που αποτελούν μέτρο δυσκολίας του στιγμιότυπου και σαν συνάρτηση των οποίων **εκφράζουμε** την πολυπλοκότητα.
  π.χ. n **κορυφές** γραφήματος, m **ακμές** γραφήματος.



- **Ανάλυση**: **Απόδειξη ορθότητας** και **εκτίμηση** πολυπλοκότητας (χρόνος n, $n \log n$, n^2).
  Ανάλυση **χειρότερης**, **μέσης**, και **καλύτερης** περίπτωσης.
- Ανάλυση καθορίζει καταλληλότερη λύση ανάλογα με **απαιτήσεις εφαρμογής**.

Πειραματική Μελέτη

- Υλοποίηση αλγορίθμου σε **πρόγραμμα**.
- Δημιουργία **στιγμιοτύπων** διαφορετικού μεγέθους και **σύνθεσης**.
- Έλεγχος ορθότητας και **καταγραφή πόρων** για κάθε εκτέλεση.
- **Απεικόνιση** αποτελεσμάτων σε γραφική παράσταση και **εξαγωγή συμπερασμάτων**.



- **Περιορισμοί - Δυσκολίες**
  - Υλοποίηση χρονοβόρα και ενδεχομένως δύσκολη.
  - Αποτελέσματα όχι αντιπροσωπευτικά για άλλα στιγμιότυπα.
  - Σύγκριση υποθέτει ίδια υπολογιστικά περιβάλλοντα.

Θεωρητική Ανάλυση

- Δεν απαιτεί υλοποίηση αλλά (σαφή) **περιγραφή** του αλγορίθμου.
- Λαμβάνει υπόψη **όλα** τα στιγμιότυπα.
- Αποδεικνύει **ορθότητα**.
- Δίνει υπολογιστικούς πόρους σαν **συνάρτηση του μεγέθους** n της εισόδου (χειρότερη – μέση περίπτωση).
- Υπολογίζει **στοιχειώδεις** ανάγκες σε υπολογιστικούς πόρους.
  Ασυμπτωτική εκτίμηση: **Ανεξάρτητη** του υπολογιστικού περιβάλλοντος.
- Ανάγκες σε πόρους που εξαρτώνται από αλγόριθμο και όχι άλλους παράγοντες.
  π.χ. αρχιτεκτονική, λειτουργικό σύστημα, compiler, φορτίο υπολογιστικού συστήματος.
- Αποτελέσματα **επιβεβαιώνονται** εύκολα.
- **Μαθηματικό υπόβαθρο**: Διακριτά μαθηματικά (συνδυαστική, αθροίσματα, αναδρομικές σχέσεις, γραφήματα, μαθηματική λογική), πιθανότητες, ...

Υπολογιστικό Μοντέλο

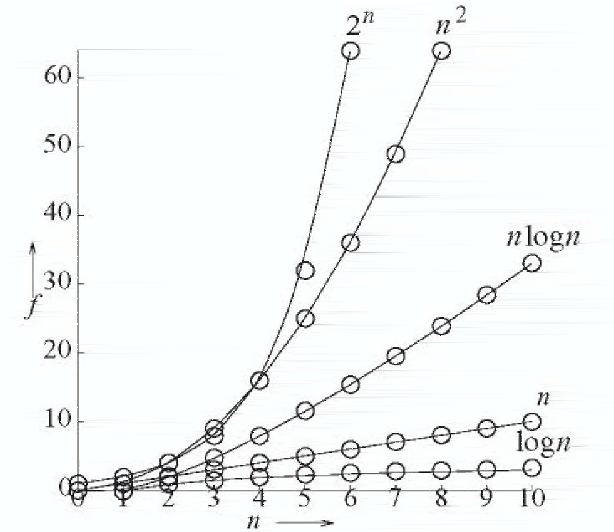
- **Μηχανή Άμεσης Προσπέλασης Μνήμης** (Random Access Machine, RAM).
  Ιδεατό μονο-επεξεργαστικό σύστημα.
- Ένας επεξεργαστής, **ακολουθιακή** εκτέλεση εντολών.
- **Απεριόριστες** θέσεις μνήμης που προσπελαύνονται σε **μοναδιαίο χρόνο**.
- **Στοιχειώδη** υπολογιστικά βήματα εκτελούνται σε **μοναδιαίο χρόνο**.
  Ανάγνωση / εγγραφή στη μνήμη, αριθμητικές και λογικές πράξεις, συγκρίσεις, εντολές ελέγχου ροής.
- Απόδειξη **ορθότητας** και προσδιορισμός **υπολογιστικών πόρων**.

Ασυμπτωτική Εκτίμηση Χρόνου Εκτέλεσης

- Σε πόσο **χρόνο** ολοκληρώνεται ο **αλγόριθμος** όταν εφαρμόζεται σε στιγμιότυπο **μεγέθους** n;
  Το αποτέλεσμα είναι μια (αύξουσα) συνάρτηση του n.
- Ενδιαφέρει **τάξη μεγέθους** χρόνου εκτέλεσης και όχι **ακριβής εκτίμηση** (συχνά δύσκολο να γίνει).
- **Ασυμπτωτική εκτίμηση** αγνοεί **σταθερές** και εστιάζει σε **τάξη μεγέθους** χρόνου εκτέλεσης **συναρτήσει του** n.
- Σταθερές εξαρτώνται από υπολογιστή, υλοποίηση, κλπ.
- Τάξη μεγέθους είναι **εγγενής ιδιότητα** του αλγόριθμου.
  max έχει **γραμμικό χρόνο** σε **όλους** τους υπολογιστές.
- Εστιάζουμε σε (πολύ) **μεγάλα στιγμιότυπα**.
  Καλύτερος αλγόριθμος $\Leftrightarrow$ 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** $\sim 2^n$



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: $a_d n^d + a_{d-1} n^{d-1} + a_1 n + a_0 = \Theta(n^d)$.
- $\sum_{i=1}^n i = \Theta(n^2)$, $\sum_{i=1}^n i^2 = \Theta(n^3)$, …, $\sum_{i=1}^n i^k = \Theta(n^{k+1})$.
- $\sum_{i=1}^n 1/i = \Theta(\log n)$, $\sum_{i=1}^n 2^i = \Theta(2^n)$, $\log(n!) = \Theta(n \log n)$.

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 $(n' = n + \log 10)$

Ευεπίλυτα και Δυσεπίλυτα Προβλήματα

- **Κλάση P**: προβλήματα που λύνονται σε πολυωνυμικό χρόνο.
- **Αξίωμα Cook - Karp**: **P** ταυτίζεται με ευεπίλυτα προβλήματα.
- **Υπέρ**:
  - Δεν εξαρτάται από υπολογιστικό μοντέλο!
  - Συνήθως **μικρά** πολυώνυμα (π.χ. $n, n^2, n^3, \ldots$).
  - Διπλασιασμός υπολογιστικής ισχύος $\Rightarrow$ **σημαντική αύξηση** (π.χ. $2, \sqrt{2}, 2^{1/3}, \ldots$) μεγέθους στιγμιοτύπων.
- **Κατά**:
  - **Ακραίες περιπτώσεις**: Αλγόριθμος με χρόνο n^{100} δεν είναι πρακτικός ενώ αλγόριθμος με χρόνο $2^{n/100}$ είναι!
  - **Γραμμικός Προγραμματισμός**: Simplex εκθετικός στη θεωρία αλλά ταχύτερος στην πράξη!