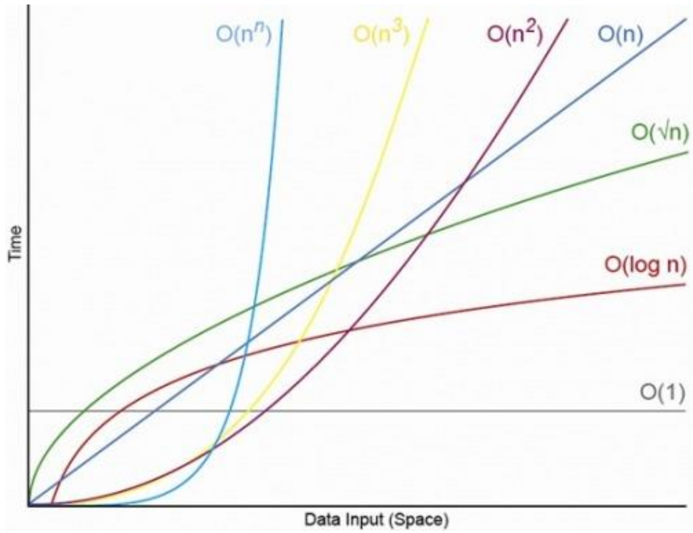
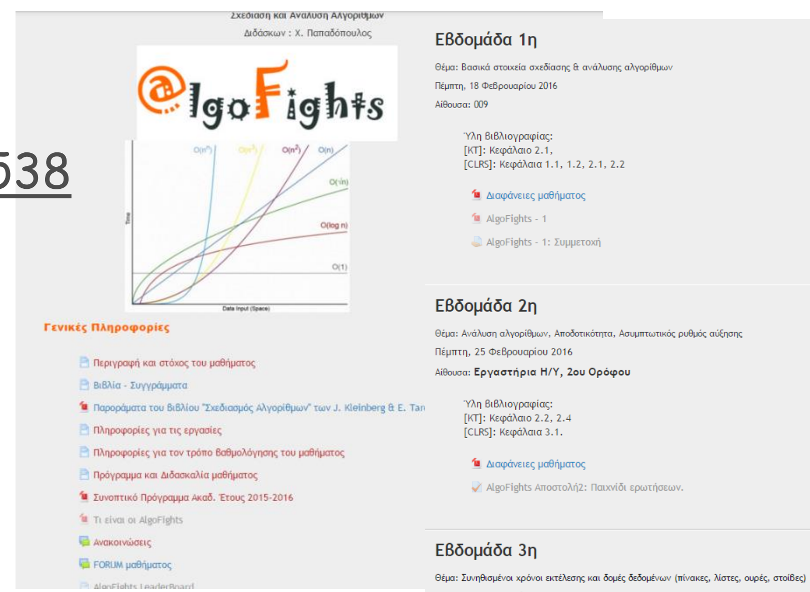
641: Σχεδίαση και Ανάλυση Αλγορίθμων

Χάρης Παπαδόπουλος



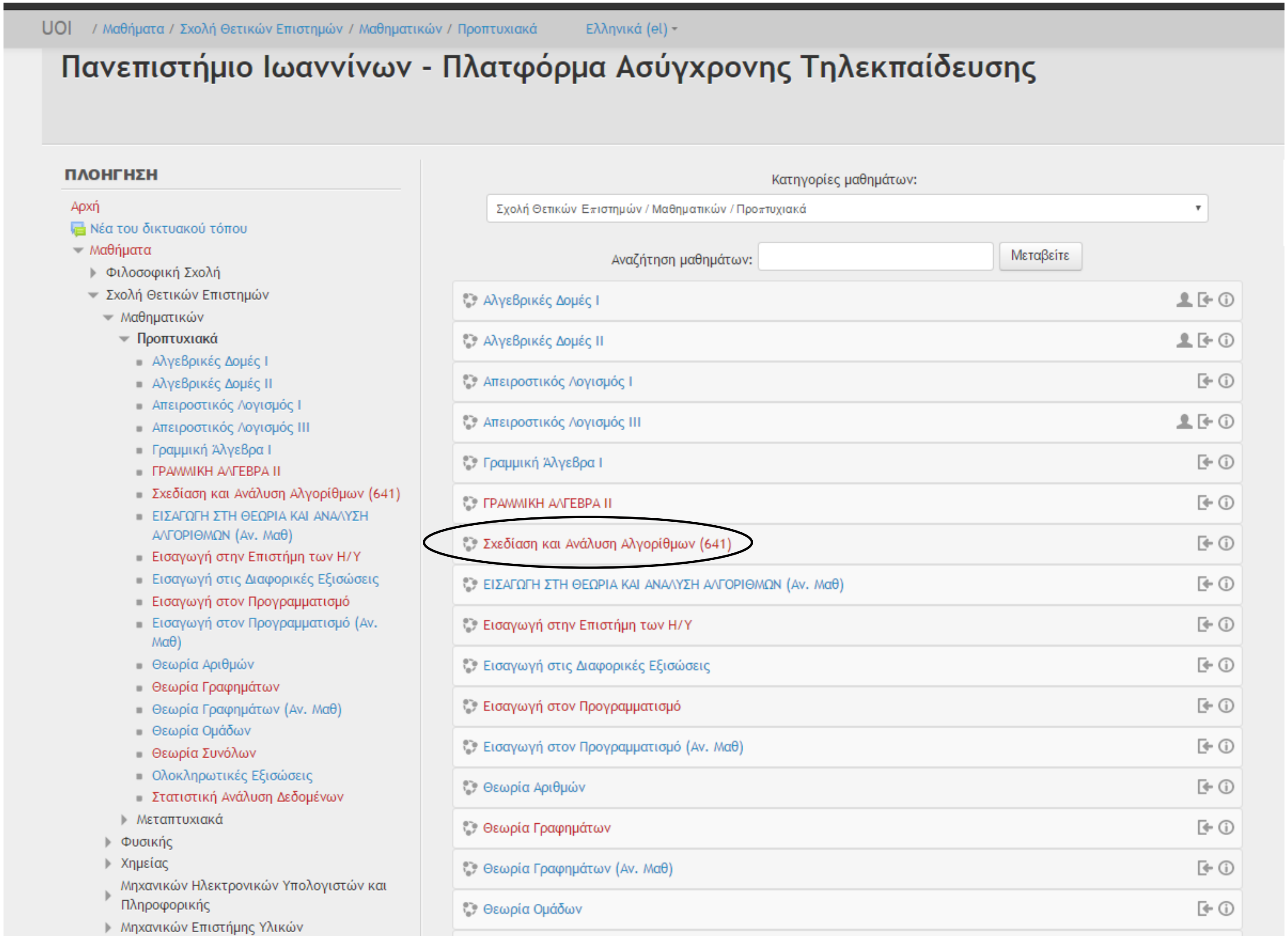
Διαδικαστικά Μαθήματος

- Διδασκαλία:
  - Τετάρτη 12:00-15:00
- Επιπλέον:
  - Ορισμένες Πέμπτες **18:00-21:00**
  - Αναπληρώσεις μαθημάτων / Συμπληρωματικές ασκήσεις
- Ώρες Γραφείου:
  - Δευτέρα 11:00-13:00 και email
- Ιστοσελίδα μαθήματος:
  - ecourse.uoi.gr/course/view.php?id=538
    - Διαφάνειες ,
    - Βιντεοδιαλέξεις,
    - Ασκήσεις,
    - Χρήσιμο υλικό,
    - Ανακοινώσεις



Διαδικαστικά Μαθήματος

- **Ιστοσελίδα μαθήματος:**
  - ecourse.uoi.gr/course/view.php?id=538



Τρόπος Εξέτασης και Ασκήσεις

- ## Γραπτές Εργασίες
  - Θα ανακοινωθούν 2 Γραπτές Εργασίες

  - Κάθε Εργασία
    - έχει βάρος 10% του τελικού βαθμού
    - έως 2 άτομα
    - **απαραίτητη** προφορική παρουσίαση με επιπλέον βάρος 5%

  - **Αυστηρές** Ημερομηνίες Ανάρτησης και Παράδοσης
    - ~ 2 εβδομάδες για την ολοκλήρωση

- ## Τελική Εξέταση
  - Χρήση χειρόγραφου φύλλου A4 (2 σελ.) με ό,τι πληροφορία θέλετε
  - Το φύλλο το παραδίδεται μαζί με το γραπτό σας.
  - Βάρος 70% του τελικού βαθμού

- ## Τελικός Βαθμός =

  10% ΓΕ1 + 5% ΓΕ1 + 10%ΓΕ2 + 5%ΓΕ2 + 70% ΤΕ + ~~@Insights~~

@lgoFights

- Παιχνίδι «Αλγοριθμομαχίες»
  - Σκοπός του @lgoFights: να αποκτήσουμε όσο περισσότερους πόντους μπορούμε
  - Διεξαγωγή του @lgoFights: ανά τακτά χρονικά διαστήματα (~ ανά εβδομαδιαία βάση)
  - Συνήθως αφορούν απλή εφαρμογή της θεωρίας
  - Ατομικές (παραδίδονται ηλεκτρονικά ή χειρόγραφα στο μάθημα)
- Η εκφώνηση θα βρίσκεται στο ecourse
  - ανακοινώνονται ~ κάθε εβδομάδα (ίσως και από την τρέχουσα!!)

Bob

Λόγοι:

- Θέλουμε η διαδικασία της μάθησης να είναι διασκεδαστική
- Ευκαιρία σε κάθε έναν να μάθ[illegible] με τον δικό του [illegible]θμό.
- Θα δοθούν πολλές και διάφ[illegible]ρίε[illegible]ελτιώσετε την πρόοδό σας αλλά και τον βαθμό σ[illegible]

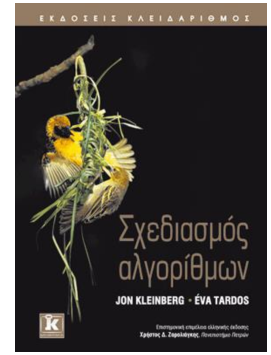
Απόλυτη επιτυχία παιχνιδιού [illegible]οντοι): μ[illegible]λέον μονάδα στο ΤΒ.

Διαλέξεις

- Διαφάνειες στο ecourse
  - Υπάρχουν ήδη διαθέσιμες οι διαφάνειες του ακ. έτος 2016-17 για να έχετε μια "εικόνα" της διάλεξης
  - Ωστόσο κάθε εβδομάδα θα αλλάζουν σύμφωνα με τις νέες διαλέξεις
- Βιντεοσκοπημένες διαλέξεις
  - Είχαν βιντεοσκοπηθεί το ακ. έτος 2012-13
  - Στα πλαίσια ενός προγράμματος ανάπτυξης («Ψηφιακές Δράσεις»)
  - Είναι διαθέσιμες στο ecourse

Βιβλιογραφία

[KT] J. Kleinberg and E. Tardos,
Σχεδιασμός Αλγορίθμων, ελληνική έκδοση,
Εκδόσεις Κλειδάριθμος, 2008
(4η επιλογή στον Εύδοξο)



[CLRS] T. Cormen, C. Leiserson, R. Rivest, and C. Stein,
Εισαγωγή στους Αλγορίθμους, ελληνική έκδοση,
Πανεπιστημιακές Εκδόσεις Κρήτης, 2012

[DPV] S. Dasgupta, C. Papadimitriou, and U. Vazirani,
Αλγόριθμοι, ελληνική έκδοση,
Εκδόσεις Κλειδάριθμος, 2008

Στόχος του Μαθήματος

- Εισαγωγή σε θεμελιώδεις αλγοριθμικές έννοιες και τεχνικές

- *Αλληγορία & Μήνυμα*

Έστω ότι θέλετε να γίνετε γλύπτης.
Χρειάζεστε **εκμάθηση βασικών τεχνικών**: εύρεση σωστών υλικών (μάρμαρα, πέτρες, κλπ), μετακίνηση υλικών, ανέγερση σκαλωσιάς, …

- Η γνώση αυτών των τεχνικών δεν θα σας κάνει διάσημο καλλιτέχνη, αλλά ακόμη και αν έχετε ένα πραγματικά εξαιρετικό ταλέντο θα είναι πολύ δύσκολο να γίνετε επιτυχημένος καλλιτέχνης, χωρίς να τις γνωρίζετε.

- Δεν είναι αναγκαίο να κατέχετε τέλεια όλες τις βασικές τεχνικές πριν φτιάξετε το πρώτο σας γλυπτό. Αλλά πρέπει πάντοτε να είστε πρόθυμοι να επιστρέψετε στις βασικές σας τεχνικές προκειμένου να τις βελτιώσετε.

Αλγόριθμος - Ορισμός

- [webster.com] Μια διαδικασία για την επίλυση ενός μαθηματικού προβλήματος σε ένα πεπερασμένο αριθμό βημάτων που συχνά περιέχει την επανάληψη μιας πράξης.

- [Knuth] Ένας αλγόριθμος είναι μια πεπερασμένη, συγκεκριμένη, αποτελεσματική διαδικασία, με μια είσοδο και κάποια έξοδο.

> Οι σπουδαίοι αλγόριθμοι είναι η ποίηση του υπολογισμού. Όπως μια στροφή ενός ποιήματος, μπορεί να είναι λακωνικοί, υπαινικτικοί, πυκνοί, ακόμα και μυστηριώδεις. Όταν όμως ξεκλειδωθούν, ρίχνουν ένα λαμπρό φως σε κάποια διάσταση, του υπολογισμού. - *Francis Sullivan*

Ετοιμολογία

Αλγορισμός = διαδικασία εκτέλεσης αριθμητικής χρησιμοποιώντας δεκαδικούς (αραβικούς) αριθμούς

- Λάθος αντίληψη ! Αλγόριθμος ≠ *άλγος/αλγεινός* [επίπονος] + αριθμός

Πραγματική Προέλευση: παραφθορά του ονόματος του *Abu 'Abd Allah Muhammad ibn Musa* ***al-Khwarizm*** (780 – 835/850 μΧ), Πέρση μαθηματικού, αστρονόμου και συγγραφέα διδακτικών βιβλίων από την επαρχία Khorasan (σημερινό Uzbekistan). Έγραψε το Kitab, **al-jabr** wa'l-muqabala, που εξελίχθηκε στο σημερινό εγχειρίδιο της άλγεβρας του λυκείου.

Σχεδίαση & Ανάλυση Αλγορίθμων

- Ιδέα παλιά …

- … αλλά συστηματική μελέτη και ανάλυση μετά το 1970

- Απαιτεί: κριτική μαθηματική σκέψη και επίλυση προβλημάτων

Design and analysis of algorithms (Κλασική ύλη):

- Greedy.
- Divide-and-conquer.
- Dynamic programming.
- Network flow.
- Randomized algorithms.
- Intractability.
- Coping with intractability.

Παράδειγμα

Έστω 12 μπίλιες εκ των οποίων **μια μόνο** έχει διαφορετικό βάρος από τις υπόλοιπες. Η μπίλια με διαφορετικό βάρος μπορεί να είναι ελαφρύτερη ή βαρύτερη από τις υπόλοιπες.

Σας δίνεται και μια ζυγαριά σύγκρισης (πλάστιγγα) .

Χρησιμοποιώντας την ζυγαριά **όσο το δυνατό λιγότερες φορές** (π.χ. **4 ή 3 φορές)** βρείτε την παράταιρη μπίλια και να προσδιορίσετε αν η μπίλια είναι ελαφρύτερη ή βαρύτερη.

Λέξεις "κλειδιά": *ελαχιστοποίηση* και *κάτω φράγμα*

Εφαρμογές

Αλγόριθμοι: βρίσκονται στην «καρδιά» κάθε υπολογιστικής διαδικασίας

- Δίκτυα (επικοινωνιών, μεταφορών, κλπ)
- Επιχειρησιακή έρευνα
- Τεχνητή νοημοσύνη
- Υπολογιστική βιολογία
- Βάσεις δεδομένων
- Υπολογιστική Γεωμετρία
- Ανάλυση δεδομένων (κοινωνική δικτύωση)
- Επεξεργασία σημάτων
- Γραφικά υπολογιστών
- Επιστημονικός υπολογισμός
- Αριθμητική Ανάλυση
- . . .

Εστιάζουμε σε αλγόριθμους και τεχνικές που είναι χρήσιμοι στην πράξη

Υλη του μαθήματος

- Βασικά στοιχεία σχεδίασης & ανάλυσης αλγορίθμων
- Ανάλυση αλγορίθμων, αποδοτικότητα, ασυμπτωτικός συμβολισμός
- Συνηθισμένοι χρόνοι εκτέλεσης και βασικές δομές δεδομένων
  - πίνακες, λίστες, στοίβες, ουρές
- Ευσταθές ταίριασμα, ορθότητα, σωρός και ουρά προτεραιότητας
- Μέθοδος «Διαίρει και Βασίλευε»
  - Εφαρμογές σε ταξινόμηση στοιχείων
  - Επίλυση αναδρομικών σχέσεων

- Γραφήματα και αλγόριθμοι γραφημάτων
  - Διάτρεξη γραφημάτων (BFS, DFS)
  - Συνεκτικότητα
  - Τοπολογική διάταξη
- Μέθοδοι «Απληστείας» και «Δυναμικού Προγραμματισμού»
  - Ελάχιστα σκελετικά δένδρα (αλγόριθμος Prim, αλγόριθμος Kruskal)
  - Συντομότερες διαδρομές (αλγόριθμος Dijkstra, Ροή δικτύου)
  - Χρονοπρογραμματισμός
- Επιλεγμένα θέματα
  - Υπολογιστική πολυπλοκότητα, NP-πληρότητα

έντονη ημερομηνία: μάθημα

ΕΑ: Εκφώνηση Άσκησης

ΗΠ: Ημερομηνία Παράδοσης Άσκησης

ΠΑ: Παρουσίαση Άσκησης

Ημερολόγιο μαθήματος

Φεβρουάριος

ΔΕ	ΤΡ	ΤΕ	ΠΕ	ΠΑ
17	18	**19**	20	21
24	25	**26**	27	28

Μάρτιος

ΔΕ	ΤΡ	ΤΕ	ΠΕ	ΠΑ
2	3	**4**	5	6
9	10	**11**	12	13
16	17	**18-ΕΑ**	19	20
23	24	25	26	27
30-ΗΠ	31			

Απρίλιος

ΔΕ	ΤΡ	ΤΕ	ΠΕ	ΠΑ
		1-ΠΑ	**2-ΠΑ**	3
6	7	**8**	9	10
13	14	15	16	17
20	21	22	23	24
27	28	**29**	30	

Μάιος

ΔΕ	ΤΡ	ΤΕ	ΠΕ	ΠΑ
4	5	**6**	7	8
11	12	**13-ΕΑ**	14	15
18	19	**20-ΗΠ**	21	22
25	26	27	28	29

Βασικά Στοιχεία Ανάλυσης Αλγορίθμων

- Αποδοτικότητα
- Προσεγγίσεις Ανάλυσης
- Πολυπλοκότητα

Αποδοτικότητα

- Επιθυμία: εύρεση του πιο «αποδοτικού» αλγορίθμου

- *Πώς μετράμε την αποδοτικότητα;*

- Κλιμάκωση: *πώς μεταβάλλεται η συμπεριφορά του αλγορίθμου σε διαφορετικά μεγέθη (εισόδου) του προβλήματος ;*

    - Π.χ. αναμένουμε μεγαλύτερη (υπολογιστική) προσπάθεια για μια ταξινόμηση 1.000.000 στοιχείων από ότι για μια ταξινόμηση 10.000 στοιχείων

1η προσέγγιση (εμπειρική):
προγραμματισμός αλγορίθμου και εξαγωγή συμπερασμάτων μέσω πειραμάτων

- *Ανομοιογενές* - εξαρτάται από πολλούς υποκειμενικούς παράγοντες

- Καμία γνώση για την *κλιμάκωση* (συμπεριφορά αλγορίθμου σε μεγαλύτερα μεγέθη εισόδου)

Αποδοτικότητα

Ανάλυση Αλγορίθμων:
Πρόβλεψη απαιτούμενων πόρων σαν συνάρτηση του μεγέθους της εισόδου

- Πόροι: <u>χρόνος</u>, χώρος (στη μνήμη)

- Ανάλυση: απαιτεί **μοντέλο αξιολόγησης** της *πολυπλοκότητας* ενός αλγορίθμου

- Χρονική πολυπλοκότητα (ή χρόνος εκτέλεσης): αριθμός στοιχειωδών λειτουργιών που εκτελούνται από τον αλγόριθμο σαν συνάρτηση του μεγέθους της εισόδου

2η προσέγγιση (πολυπλοκότητα χειρότερης περίπτωσης):
Άνω όριο στον χρόνο εκτέλεσης για κάθε στιγμιότυπο εισόδου

- Ομοιογενής – αποτυπώνει αρκετά καλά την αποδοτικότητα στην πράξη και προσφέρει πληροφορία κλιμάκωσης
- Δρακόντειο μέτρο, αλλά δεν υπάρχει αποτελεσματική εναλλακτική λύση

3^{η} προσέγγιση (πολυπλοκότητα μέσης περίπτωσης):
Μέσος (αναμενόμενος) χρόνος εκτέλεσης σε «τυχαία» στιγμιότυπα εισόδου μιας πιθανοτικής κατανομής

- Δύσκολο/αδύνατο να οριστούν πραγματικά στιγμιότυπα από τυχαίες κατανομές
- Ανομοιογενής - βέλτιστος αλγόριθμος για μια κατανομή μπορεί να έχει τελείως διαφορετική συμπεριφορά σε άλλη κατανομή

Αποδοτικότητα

Επικεντρωνόμαστε στην ανάλυση πολυπλοκότητας χειρότερης περίπτωσης

Μοντέλο RAM

(Μηχανή Τυχαίας Προσπέλασης – Random Access Machine)

- Στιγμιότυπο εισόδου μεγέθους n στοιχείων, με μέγιστη τιμή στοιχείου N

- Στοιχειώδες υπολογιστικό βήμα:
  - 1 μονάδα χρόνου για κάθε βασική λειτουργία (π.χ., αριθμητική πράξη, λογική πράξη, σύγκριση, καταχώρηση, κλπ)
  - 1 μονάδα χρόνου για προσπέλαση (ανάγνωση/εγγραφή) ενός σταθερού αριθμού λέξεων μνήμης των log n (ή log N) bit

- Επιθυμία κλιμάκωσης →
  - Χρονική πολυπλοκότητα = f(n) [ή f(n, log N)]
  - Χωρική πολυπλοκότητα = g(n, log N)

- Ερώτημα: *τι μορφή (πρέπει να) έχουν οι f και g ;*

Αναζήτηση Ωμής Βίας – (Υπερ)Εκθετικός Χρόνος

Ωμή βία: Τετριμμένος αλγόριθμος επίλυσης προβλήματος – έλεγχος όλων των πιθανών λύσεων (στο χώρο όλων των δυνατών λύσεων του προβλήματος)

- Συνήθως παίρνει χρόνο 2^n ή χειρότερο (π.χ. n!) για είσοδο μεγέθους n
- Μηδαμινή πρακτική αξία
- Πολύ συχνά μη επιτρεπτό (όσο μεγαλώνει η τιμή του n)

Εκθετική πολυπλοκότητα

> Υπάρχουν σταθερές a > 0, c > 0, d > 0, και k ≥ 0 τέτοιες ώστε για κάθε είσοδο μεγέθους ***n***, ο χρόνος εκτέλεσης να φράσσεται από $c \cdot a^{n^d} \log^k n$ στοιχειώδη υπολογιστικά βήματα

Π.χ., $n! \leq n^n = 2^{n \log n}$

Πολυωνυμικός Χρόνος

Υπάρχουν σταθερές $c > 0$, $d > 0$, και $k \geq 0$ τέτοιες ώστε για κάθε είσοδο μεγέθους ***n***, ο χρόνος εκτέλεσης να φράσσεται από $c \cdot n^d \cdot \log^k n$ στοιχειώδη υπολογιστικά βήματα

Επιθυμητή ιδιότητα κλιμάκωσης. Όταν το μέγεθος της εισόδου διπλασιάζεται, η χρονική πολυπλοκότητα θα πρέπει να μεγαλώνει μόνο κατά έναν σταθερό παράγοντα **Δ**.

Μέγεθος	Χρόνος
n	$c \cdot n^d \cdot \log^k n$
$2n$	$c \cdot (2n)^d \cdot \log^k (2n) <$ $c \cdot 2^{d+1} \cdot n^d \cdot \log^k n$

$\Delta = 2^{d+1}$

Ένας αλγόριθμος είναι πολυωνυμικού χρόνου αν ισχύει η παραπάνω ιδιότητα κλιμάκωσης.

Πολυωνυμικός vs Εκθετικός Χρόνος

Χρονική διάρκεια 1 ώρας σε Η/Υ με S_1 στοιχ. λειτουργίες /sec

Αλγόριθμος	ΧΠ	Μέγεθος στιγμυοτύπου
Πολυωνυμικός	n^2	n_1
Εκθετικός	2^n	N_1

- Νέος Η/Υ με ταχύτητα $S_2 = 100 \cdot S_1$ στοιχ. λειτουργίες/sec

Ερώτημα 1: πόσο μεγαλύτερα στιγμιότυπα εισόδου μεγέθους n_2 μπορεί να επιλύσει σε μία ώρα ο Π στο νέο Η/Υ ??

$$\left(\frac{n_2}{n_1}\right)^2 = \frac{s_2}{s_1} \Rightarrow \frac{n_2}{n_1} = \sqrt{\frac{s_2}{s_1}} = \sqrt{100} = 10 \Rightarrow n_2 = 10n_1$$

Ερώτημα 2: πόσο μεγαλύτερα στιγμιότυπα εισόδου μεγέθους N_2 μπορεί να επιλύσει σε μία ώρα ο Ε στο νέο Η/Υ ??

$$\frac{2^{N_2}}{2^{N_1}} = \frac{s_2}{s_1} \Leftrightarrow 2^{N_2 - N_1} = \frac{s_2}{s_1} = 100 \Leftrightarrow N_2 - N_1 = \log(100) \quad < \quad 7 \Leftrightarrow$$

$$N_2 \quad < \quad N_1 + 7 \ \textbf{(!)}$$

Ενδεικτικοί Χρόνοι Εκτέλεσης Πολυωνυμικών & Εκθετικών Αλγορίθμων

Πίνακας 2.1 Οι χρόνοι εκτέλεσης (στρογγυλεμένοι προς τα επάνω) διαφόρων αλγορίθμων για εισόδους αυξανόμενου μεγέθους, σε έναν επεξεργαστή που εκτελεί ένα εκατομμύριο εντολές υψηλού επιπέδου ανά δευτερόλεπτο. Στις περιπτώσεις όπου ο χρόνος εκτέλεσης υπερβαίνει τα 10^{25} έτη, καταγράφουμε απλώς ότι ο αλγόριθμος χρειάζεται πάρα πολύ χρόνο.

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} έτη
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 λεπτά	36 έτη	πάρα πολύ
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 έτη	10^{17} έτη	πάρα πολύ
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 λεπτά	πάρα πολύ	πάρα πολύ	πάρα πολύ
$n = 10{,}000$	< 1 sec	< 1 sec	2 λεπτά	12 ημέρες	πάρα πολύ	πάρα πολύ	πάρα πολύ
$n = 100{,}000$	< 1 sec	2 sec	3 ώρες	32 έτη	πάρα πολύ	πάρα πολύ	πάρα πολύ
$n = 1{,}000{,}000$	1 sec	20 sec	12 ημέρες	31,710 έτη	πάρα πολύ	πάρα πολύ	πάρα πολύ

Αποδοτικότητα & Πολυωνυμικός Χρόνος

Ένας αλγόριθμος είναι αποδοτικός αν έχει πολυωνυμικό χρόνο εκτέλεσης

Αιτιολόγηση: Μεγάλη πρακτική σημασία !

- Στην πράξη, οι αλγόριθμοι πολυωνυμικού χρόνου που αναπτύσσουμε σχεδόν πάντα έχουν χαμηλές σταθερές και χαμηλούς εκθέτες.
- Μείωση του εκθετικού ορίου ενός αλγόριθμου ωμής βίας συνήθως αποκαλύπτει κάποια πολύ σημαντική δομή του προβλήματος.

Εξαιρέσεις

- Κάποιοι αλγόριθμοι πολυωνυμικού χρόνου έχουν όντως υψηλές σταθερές ή/και εκθέτες και είναι άχρηστοι στην πράξη.
- Ορισμένοι εκθετικού χρόνου (ή χειρότεροι) αλγόριθμοι χρησιμοποιούνται ευρέως γιατί τα στιγμιότυπα χειρότερης περίπτωσης είναι σπάνια.

simplex method
Unix grep

Πολυωνυμικός Χρόνος

Ένας αλγόριθμος είναι **αποδοτικός** αν έχει πολυωνυμικό χρόνο εκτέλεσης

Ερώτημα: *Μας αρκεί οποιοσδήποτε πολυωνυμικός χρόνος;*

Πολυωνυμικοί Αλγόριθμοι

Πρόβλημα: ταξινόμηση n στοιχείων

Στοιχειώδης λειτουργία: σύγκριση δύο στοιχείων

Αλγόριθμος	Χρόνος (συγκρίσεις)
A1	$10 \cdot n \cdot \log n$
A2	n^2

Υπολογιστής	Συγκρίσεις/sec
Υ1 (2005)	10^7
Υ2 (2014)	10^9

$n = 10^6$

- Χρόνος A1 (πιο αποδοτικός) στον Υ1 (αργός):

  $(10 \cdot 10^6 \cdot \log 10^6) / 10^7 = 6 \cdot \log 10$ secs ≈ 20 **secs**

- Χρόνος A2 (λιγότερο αποδοτικός) στον Υ2 (γρήγορος):

  $(10^6)^2 / 10^9 = 1000$ secs = 16.6 **min**

Πολυωνυμικοί Αλγόριθμοι

Πρόβλημα: ταξινόμηση n στοιχείων

Στοιχειώδης λειτουργία: σύγκριση δύο στοιχείων

Αλγόριθμος	Χρόνος (συγκρίσεις)
A1	$10 \cdot n \cdot \log n$
A2	n^2

Υπολογιστής	Συγκρίσεις/sec
Y1 (2005)	10^7
Y2 (2014)	10^9

$n = 10^7$

- Χρόνος A1 (πιο αποδοτικός) στον Y1 (αργός):

  $(10 \cdot 10^7 \cdot \log 10^7) / 10^7 = 70 \cdot \log 10 = 232$ secs ≈ 3.8 **min**

- Χρόνος A2 (λιγότερο αποδοτικός) στον Y2 (γρήγορος):

  $(10^7)^2 / 10^9 = 10000$ secs = 27.7 **hours (!)**

Πολυωνυμικός Χρόνος

Ένας αλγόριθμος είναι αποδοτικός αν έχει πολυωνυμικό χρόνο εκτέλεσης

Ερώτημα: *Μας αρκεί οποιοσδήποτε πολυωνυμικός χρόνος;*

Απάντηση: ΟΧΙ

Αποδοτικότητα ≡

- Όσο το δυνατόν «μικρότερο» πολυώνυμο
- Τεράστια πρακτική/τεχνολογική σημασία

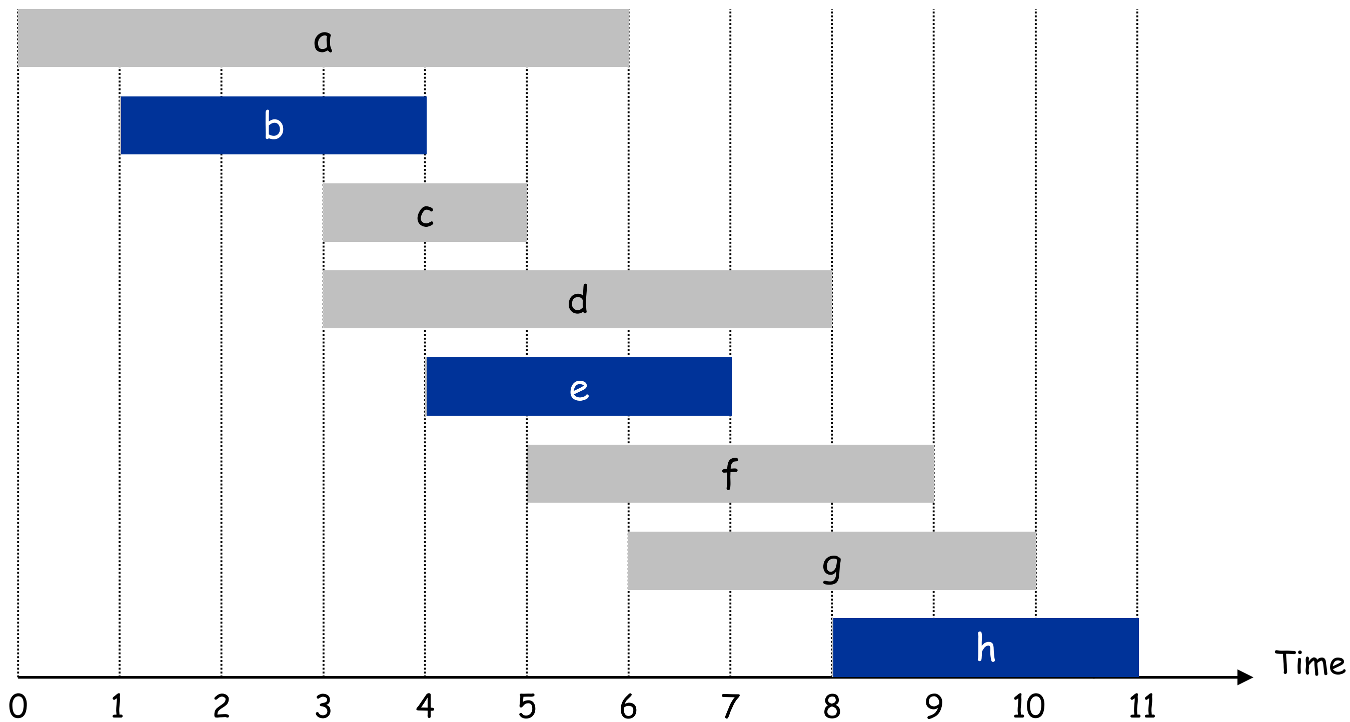
Ορισμένα Αντιπροσωπευτικά Προβλήματα

Χρονοπρογραμματισμός Διαστημάτων

Είσοδος. Σύνολο από εργασίες με χρόνους έναρξης και τερματισμού.

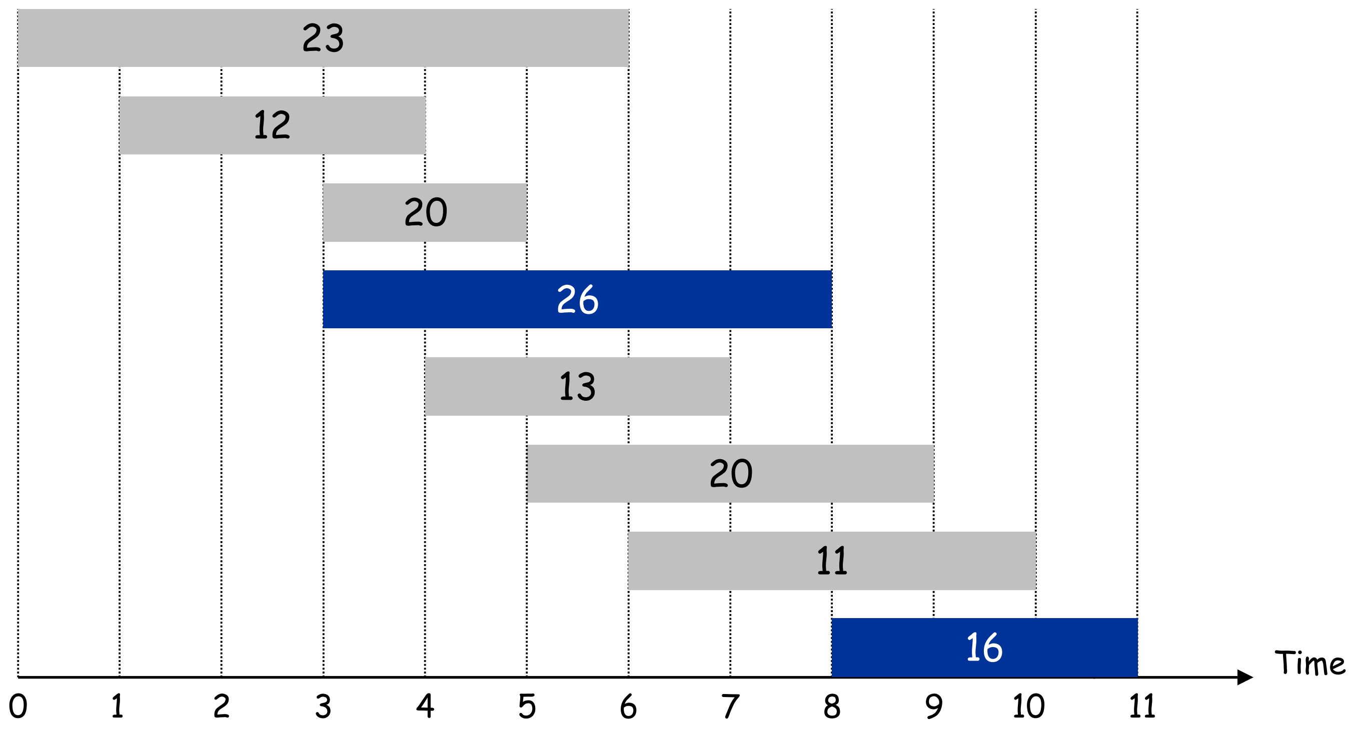
Στόχος. Εύρεση υποσυνόλου μέγιστης πληθικότητας από συμβατές εργασίες.

οι εργασίες δεν επικαλύπτονται



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

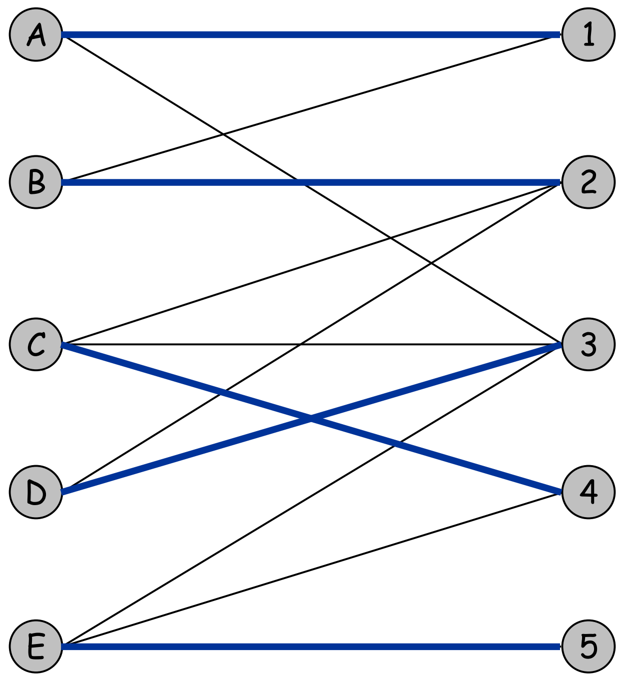
Είσοδος. Σύνολο από εργασίες με χρόνους έναρξης και τερματισμού και βάρη.
Στόχος. Εύρεση υποσυνόλου από συμβατές εργασίες με μέγιστο συνολικό βάρος.



Διμερές Ταίριασμα

Είσοδος. Ένα διμερές γράφημα.

Στόχος. Εύρεση ταιριάσματος μέγιστης πληθικότητας.

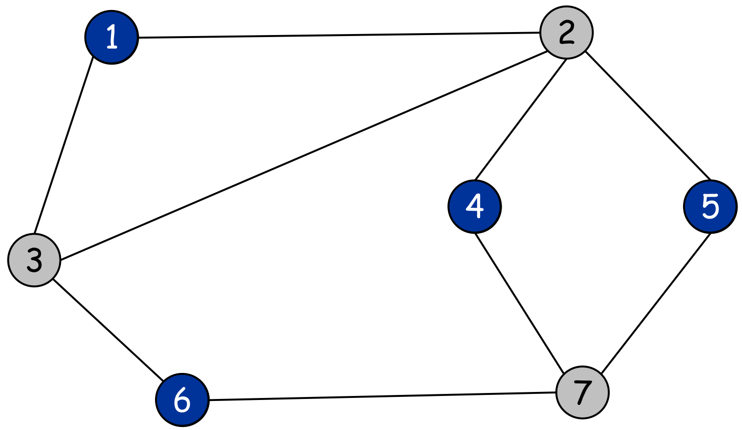


Ανεξάρτητο Σύνολο

Είσοδος. Ένα γράφημα.

Στόχος. Εύρεση ανεξάρτητου συνόλου μέγιστης πληθικότητας.

υποσύνολο κορυφών που ανά δύο δεν ενώνονται με ακμή



Ορισμένα Αντιπροσωπευτικά Προβλήματα

Παραλλαγές του θέματος: ανεξάρτητο σύνολο.

Χρονοπρογραμματισμός διαστημάτων:

n log n - άπληστος αλγόριθμος.

Σταθμισμένος χρονοπρογραμματισμός διαστημάτων:

n log n - αλγόριθμος με δυναμικό προγραμματισμό.

Διμερές ταίριασμα:

n^k - αλγόριθμος ροής δικτύου.

Ανεξάρτητο σύνολο:

NP-πλήρες