Αλγόριθμοι και Πολυπλοκότητα

7ο εξάμηνο ΣΗΜΜΥ

Εισαγωγή

Διδάσκοντες: Άρης Παγουρτζής, Δώρα Σούλιου

Στάθης Ζάχος, Δημήτρης Σακαβάλας

Επιμέλεια διαφανειών: Άρης Παγουρτζής

www.corelab.ntua.gr/courses/algorithms

Αλγόριθοι και Πολυπλοκότητα

- Αυτά τα ξέρω! (sorting, Dijkstra, bfs/dfs, δέντρα, Prim, Kruskal…)
- Τα ξέρω;; (P=?NP, καλύτερος αλγόριθμος πολλαπλασιασμού, επίλυση TSP, έλεγχος πρώτων, streaming algorithms ;;)
- Επιστημονική περιοχή διαρκώς αναπτυσσόμενη, με τεράστιο εύρος και βάθος. Πολλά αναπάντητα ερωτήματα!

Κεντρικό ερώτημα

Τι μπορούμε να κάνουμε με υπολογιστή, **πώς**, και **πόσο καλά**;

Κεντρικό ερώτημα

Υπολογισιμότητα

Τι μπορούμε να κάνουμε με υπολογιστή, **πώς**, και **πόσο καλά**;

Κεντρικό ερώτημα

Υπολογισιμότητα

Τι μπορούμε να κάνουμε με υπολογιστή, **πώς**, και **πόσο καλά**;

Αλγόριθμοι

Κεντρικό ερώτημα

Υπολογισιμότητα

Τι μπορούμε να κάνουμε με υπολογιστή, **πώς**, και **πόσο καλά**;

Αλγόριθμοι

Πολυπλοκότητα

Αλγόριθμος

- Αυστηρά καθορισμένη σειρά βημάτων που επενεργεί σε δεδομένα εισόδου και παράγει δεδομένα εξόδου, σε πεπερασμένο χρόνο.
- Λύνει πρόβλημα Π αν για κάθε στιγμιότυπο παράγει την σωστή έξοδο.
- Μπορεί να εκτελεστεί «με μολύβι και χαρτί» (*μηχανιστικά αποτελεσματικός*).
- Ανάλυση αλγορίθμου: ορθότητα (απόδειξη ή αντ/δειγμα), πολυπλοκότητα.

Αμπού Αμπντουλάχ Μοχάμεντ ιμπν Μουσά αλ-Χουαρίζμι

أبو عبد الله محمد بن موسى الخوارزمي

(781 – 850 μΧ)

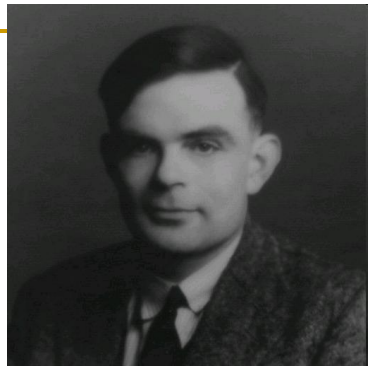
(Υπολογιστική) πολυπλοκότητα

- Αλγορίθμου: το κόστος του αλγορίθμου σε χρόνο, χώρο (μνήμη), επεξεργαστές (παραλληλία), επικοινωνία (bandwidth), ενέργεια (sensors, robots),...
- Προβλήματος: το κόστος επίλυσης του προβλήματος σε χρόνο, χώρο, κ.λπ.
  - αν ο αλγόριθμος Α λύνει το πρόβλημα Π, τότε η πολυπλοκότητα του Π είναι το πολύ όση η πολυπλοκότητα του Α (άνω φράγμα).
  - για κάτω φράγμα πολυπλοκότητας προβλήματος χρειαζόμαστε απόδειξη.

Είδη πολυπλοκότητας

- Χειρότερης περίπτωσης (worst case): με αυτήν ασχολούμαστε συνήθως.
- Μέσης περίπτωσης (average case): με βάση κατανομή πιθανότητας στιγμιοτύπων (instances) του προβλήματος. Συνήθως δύσκολο να οριστεί σωστά.
- Καλύτερης περίπτωσης (best case): χρήσιμη εναντίον αντιπάλου.
- Αποσβετική (amortized): εκφράζει την μέση αποδοτικότητα σε μια σειρά επαναλήψεων του αλγορίθμου.

Θεωρητικές θεμελιώσεις

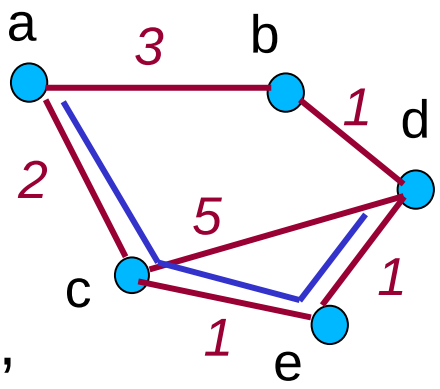


- Υπολογιστικό πρόβλημα: ορίζεται βάσει μιας επιθυμητής σχέσης εισόδου-εξόδου
  - τυπικά: διμελής σχέση μεταξύ *λέξεων* (συμβολοσειρών) από ένα *αλφάβητο*
  - η είσοδος λέγεται και στιγμιότυπο (instance), η έξοδος απάντηση ή λύση
- Αλγόριθμος για πρόβλημα: αυστηρά καθορισμένη διαδικασία που «υλοποιεί» την αντιστοίχιση:
  - για κάθε έγκυρο στιγμιότυπο υπολογίζει τη σωστή απάντηση (ή μία από τις σωστές)
  - μοντέλα: μηχανή Turing, RAM, WHILE-programs, …

Υπολογιστικά προβλήματα

Παραδείγματα

- Έλεγχος πρώτων αριθμών
  - $2^{43112609}-1$ → «ΝΑΙ»
  - 129 → «ΟΧΙ»



- Συντομότερα μονοπάτια
  - (({a,b},3), ({a,c},2), ({b,d},1), ({c,d},5), ({c,e},1), ({d,e},1), a, d) → (a,c,e,d) ή (a,b,d)

Υπολογιστικά προβλήματα

Άλλα παραδείγματα

- Πρόβλημα τερματισμού
- Πρόβλημα Collatz
- Κύκλος Euler
- Κύκλος Hamilton
- Αριθμοί Fibonacci

Πρόβλημα Τερματισμού

- Πρόβλημα Τερματισμού (Halting Problem):

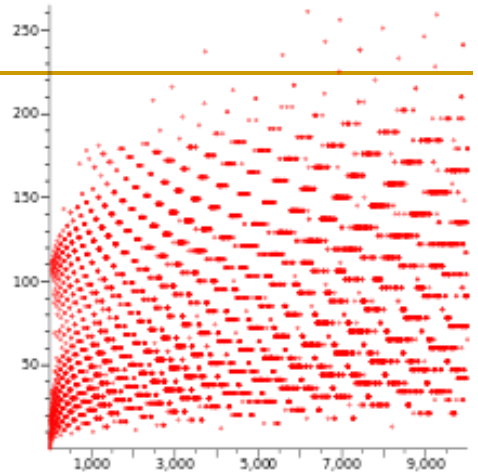
  Δίνεται πρόγραμμα και είσοδος. Σταματάει το πρόγραμμα για αυτή την είσοδο (ή "τρέχει" επ' άπειρον);

- Ισοδύναμη παραλλαγή:

  Δίνεται πρόγραμμα χωρίς είσοδο. Σταματάει;

- Δεν υπάρχει αλγόριθμος που να απαντάει σωστά σε κάθε στιγμιότυπο του Halting Problem: μη επιλύσιμο πρόβλημα !

Πρόβλημα Collatz



- Έστω το πρόγραμμα

```
while n!=1 do
   if (n is even) then n = n /2 else n =3n +1
```

- Πρόβλημα Collatz:

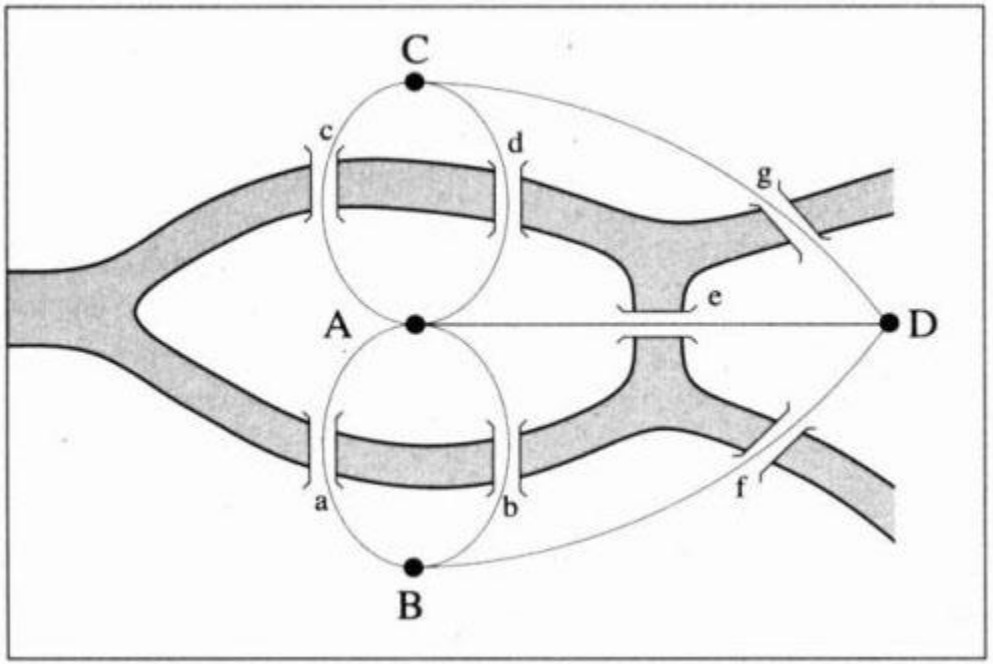
  Δίνεται φυσικός αριθμός *n*. Σταματάει το παραπάνω πρόγραμμα για είσοδο *n*;

- Παράδειγμα: `7 -> 22 -> 11 -> 34 -> 17 -> 52 -> 26 -> 13 -> 40 -> 20 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1`

- Δεν ξέρουμε αν είναι επιλύσιμο !

Κύκλος Euler

Δίνεται γράφος. Υπάρχει διαδρομή που περνάει από κάθε ακμή μια ακριβώς φορά;

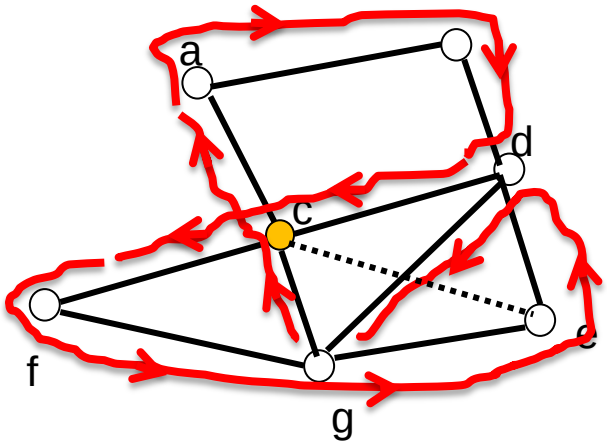


Seven Bridges of Königsberg

Source:
http://physics.weber.edu/carroll/honors_images/BarbasiBridges.jpg

Επίλυση Κύκλου Euler

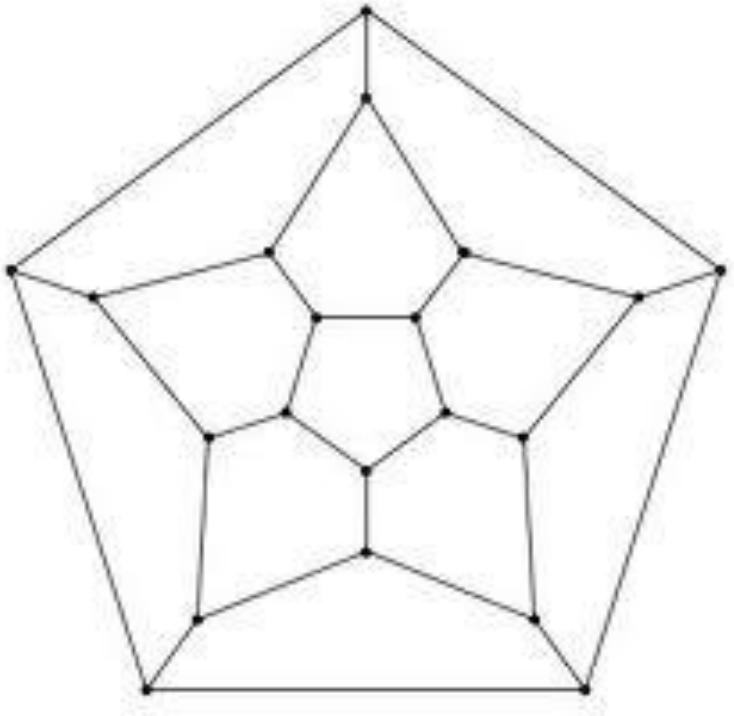
- Το πρόβλημα ΚΥΚΛΟΣ EULER είναι ευεπίλυτο.
- Η απάντηση είναι 'ΝΑΙ' ανν
  κάθε κόμβος έχει άρτιο # γειτόνων



- Για κάθε γράφο με n κόμβους αρκούν n^2 έλεγχοι: χρόνος *πολυωνυμικός* ως προς το μέγεθος της εισόδου.
- Τέτοια προβλήματα που η επίλυσή τους χρειάζεται χρόνο $O(n)$, $O(n^2)$, $O(n^3)$ … ανήκουν στην **κλάση P** (Polynomial time).

Κύκλος Hamilton

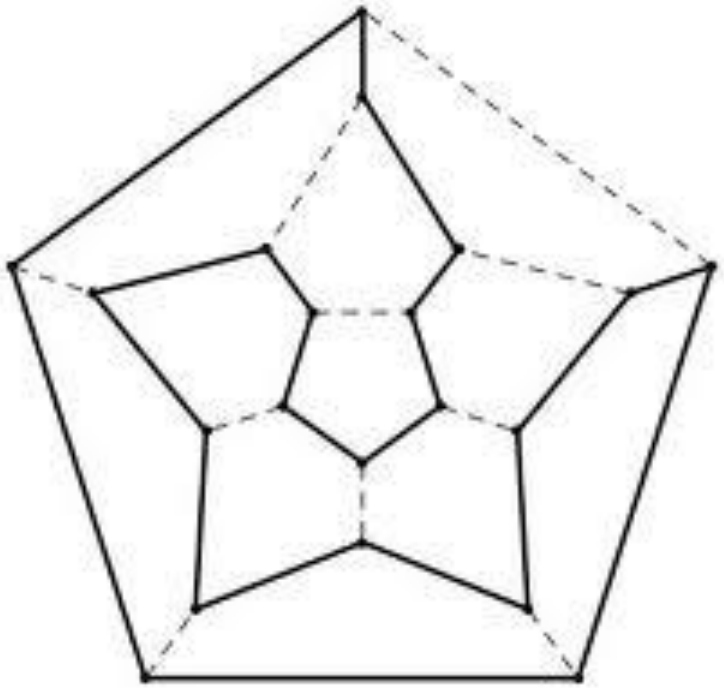
Δίνεται γράφος. Υπάρχει διαδρομή που περνάει από κάθε κορυφή μια ακριβώς φορά;



Source: http://jwilson.coe.uga.edu/emat6680/yamaguchi/emat6690/essay1/gt.html

Κύκλος Hamilton

Δίνεται γράφος. Υπάρχει διαδρομή που περνάει από κάθε κορυφή μια ακριβώς φορά;



Source: http://jwilson.coe.uga.edu/emat6680/yamaguchi/emat6690/essay1/gt.html

Επίλυση Κύκλου Hamilton

- Το πρόβλημα ΚΥΚΛΟΣ HAMILTON είναι «δύσκολο» (δυσεπίλυτο). Δεν γνωρίζουμε γρήγορο αλγόριθμο. Καλύτερος γνωστός αλγόριθμος: περίπου εξαντλητική αναζήτηση ($n!$ μεταθέσεις). Αν μας προτείνουν λύση την επαληθεύουμε πολύ γρήγορα.

- Προβλήματα που η επαλήθευση μιας λύσης (αν υπάρχει και μας δοθεί) γίνεται σε χρόνο $O(n)$, $O(n^2)$, $O(n^3)$, …, ανήκουν στην **κλάση NP** (Non-deterministic Polynomial time).

P =? NP

- Μπορεί να λυθεί το πρόβλημα του Hamilton τόσο γρήγορα όσο και το πρόβλημα του Euler;
- Αυτό είναι ουσιαστικά το **P =? NP** πρόβλημα, που αποτελεί το πιο σημαντικό ανοικτό πρόβλημα της Θεωρητικής Πληροφορικής σήμερα.
- Στο http://www.claymath.org προσφέρονται 1εκ. δολάρια για τη λύση του !

Αριθμοί Fibonacci

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

$$F_0 = 0,\ F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2},\ n >= 2$$

- Πρόβλημα Fibonacci:
  Δίνεται n, να υπολογιστεί το F_n

- Προσοχή: *δεν ανήκει στο* **P ούτε στο NP**! (*γιατί*;)

Το πρόβλημα του 2^{29}

Ποιο ψηφίο λείπει από τον αριθμό 2^{29} ;

(γνωρίζοντας ότι αποτελείται από 9 διαφορετικά ψηφία)

Το πρόβλημα του 2^{29}

Μπορείτε να βρείτε έναν αλγόριθμο;

Είναι «καλός»; Πόσες πράξεις θα κάνει;

Γίνεται καλύτερα;

Το πρόβλημα του 2^{29}

Γίνεται χωρίς να υπολογίσουμε τον αριθμό;

[*ερώτηση από βιβλίο προετοιμασίας για συνεντεύξεις για* 'quant jobs']

Κατηγορίες υπολογ. προβλημάτων

- Προβλήματα απόφασης: σε κάθε στιγμιότυπο αντιστοιχεί μία απάντηση ΝΑΙ ή ΟΧΙ.
  - Έλεγχος πρώτων, Προσβασιμότητα (Reachability), Κύκλος Hamilton
  - Η απλούστερη μορφή, όμως αρκετά ισχυρή.
  - Σημαντικές κλάσεις: **P**, **NP, PSPACE**

Κατηγορίες υπολογ. προβλημάτων

- Προβλήματα αναζήτησης (ή εύρεσης): σε κάθε στιγμιότυπο αντιστοιχεί ένα σύνολο *έγκυρων λύσεων* και ζητείται μία (οποιαδήποτε) από αυτές.
  - Τέλειο Ταίριασμα (Perfect Matching), Satisfiability, Κύκλος Hamilton (εύρεση).
  - Ειδική περίπτωση: function problems (για κάθε στιγμιότυπο μοναδική λύση) : Ταξινόμηση, Ύψωση σε Δύναμη, Παραγοντοποίηση.
  - Σημαντικές κλάσεις: **FP**, **FNP, TFNP, FPSPACE**

Κατηγορίες υπολογ. προβλημάτων

- Προβλήματα βελτιστοποίησης: ορίζονται με βάση μία *αντικειμενική συνάρτηση*. Σε κάθε στιγμιότυπο αντιστοιχεί ένα μη κενό σύνολο *αποδεκτών λύσεων* (feasible solutions) και ζητείται μία που να βελτιστοποιεί την αντικειμενική συνάρτηση.
  - Συντομότερα μονοπάτια, Μέγιστο Ταίριασμα (Maximum Matching), Travelling Salesman Problem (TSP).
  - Σημαντικές κλάσεις: **PO**, **NPO, APX**

Τυπική περιγραφή προβλημάτων

- Προβλήματα απόφασης: λογική συνάρτηση (Boolean function) – ισοδύναμα, τυπική γλώσσα (σύνολο λέξεων από αλφάβητο)
- Προβλήματα αναζήτησης / εύρεσης: διμελής σχέση που ορίζει τις έγκυρες λύσεις.
- Προβλήματα βελτιστοποίησης: διμελής σχέση που ορίζει τις αποδεκτές λύσεις και αντικειμενική συνάρτηση.

Απλοποιήσεις προβλημάτων

- Πρόβλημα αναζήτησης → πρόβλημα απόφασης: για το δοσμένο στιγμιότυπο, υπάρχει λύση;
- Πρόβλημα βελτιστοποίησης → πρόβλημα απόφασης:
  - στιγμιότυπο επαυξάνεται με τιμή *k*, ζητείται αν υπάρχει λύση με αντικειμενική τιμή το πολύ *k* (για πρόβλημα ελαχιστοποίησης) ή τουλάχιστον *k* (για πρόβλημα μεγιστοποίησης).
- Σημαντικό: συνήθως η απλοποίηση διατηρεί τη δυσκολία του προβλήματος.

Πολυπλοκότητα υπολογιστικών προβλημάτων

- Συνάρτηση κόστους επίλυσης σε χρόνο, χώρο (μνήμη), επεξεργαστές, επικοινωνία, ενέργεια,...
- Αλγόριθμοι: παρέχουν άνω φράγματα
  - ταξινόμηση (με bubblesort): $O(n^2)$
- Αποδείξεις δυσκολίας: παρέχουν κάτω φράγματα
  - ταξινόμηση με συγκρίσεις: $\Omega(n \log n)$
  - NP-πληρότητα: ισχυρή ένδειξη απουσίας αποδοτικού αλγορίθμου

Αποδοτική επίλυση προβλήματος

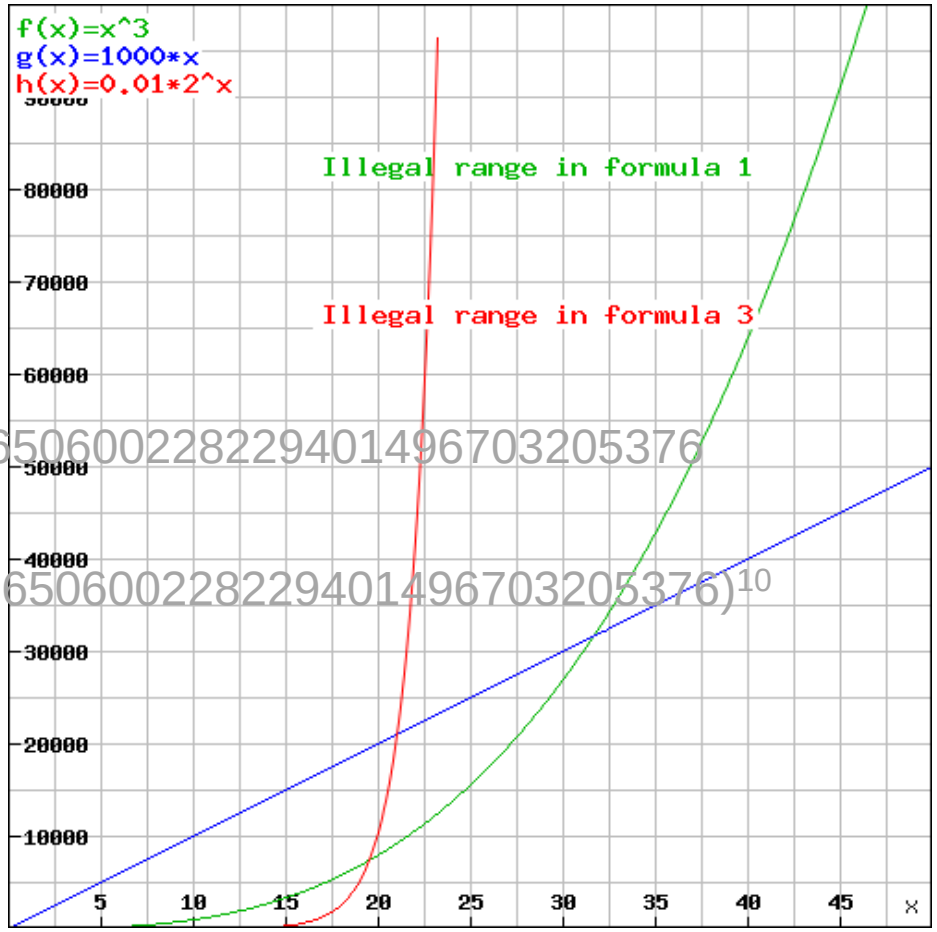
- Έχει ταυτιστεί με πολυωνυμικό χρόνο (κλάση **P**)
- Σημασία πολυωνυμικού χρόνου (πρακτικά και «χοντρικά»):

  αν μπορούμε να γράψουμε την είσοδο μπορούμε να πάρουμε και την απάντηση!!

- Παράδοξο:
  - αλγόριθμος $O(n^{100})$ θεωρείται αποδοτικός ενώ $O(2^{n/1000})$ όχι!
  - οι περιπτώσεις αυτές είναι σπάνιες

Σημασία πολυωνυμικού χρόνου

$\log n$	n	n^2	2^n
3.322	10	100	1024
6.644	100	10000	1267650600228229401496703205376
9.966	1000	1000000	$(1267650600228229401496703205376)^{10}$



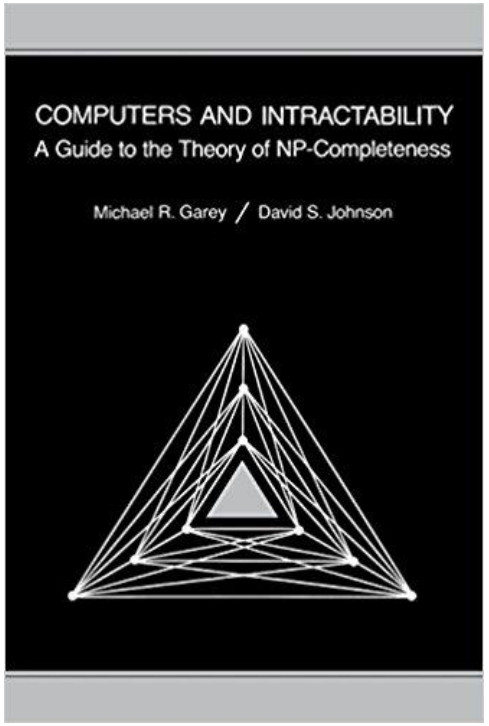
Ο ρυθμός αύξησης των εκθετικών συναρτήσεων είναι απαγορευτικός για μεγάλα στιγμιότυπα!

Αποδείξεις **NP**-πληρότητας: αναγωγές

- Κάθε πρόβλημα στο **NP** μετατρέπεται (σε πολυωνυμικό χρόνο) στο πρόβλημα Satisfiability: **NP**-*πλήρες*
- Το Satisfiability μετατρέπεται σε πολλά άλλα, π.χ. Κυκλος Hamilton
- Κυκλος Hamilton μετατρέπεται σε TSP
- Οι αναγωγές συντίθεται σε πολυωνυμικό χρόνο
- Αποτέλεσμα: όλα τα NP-πλήρη προβλήματα εξίσου δύσκολο να λυθούν σε πολυωνυμικό χρόνο, *ή είναι όλα στο* **P** *ή κανένα* !

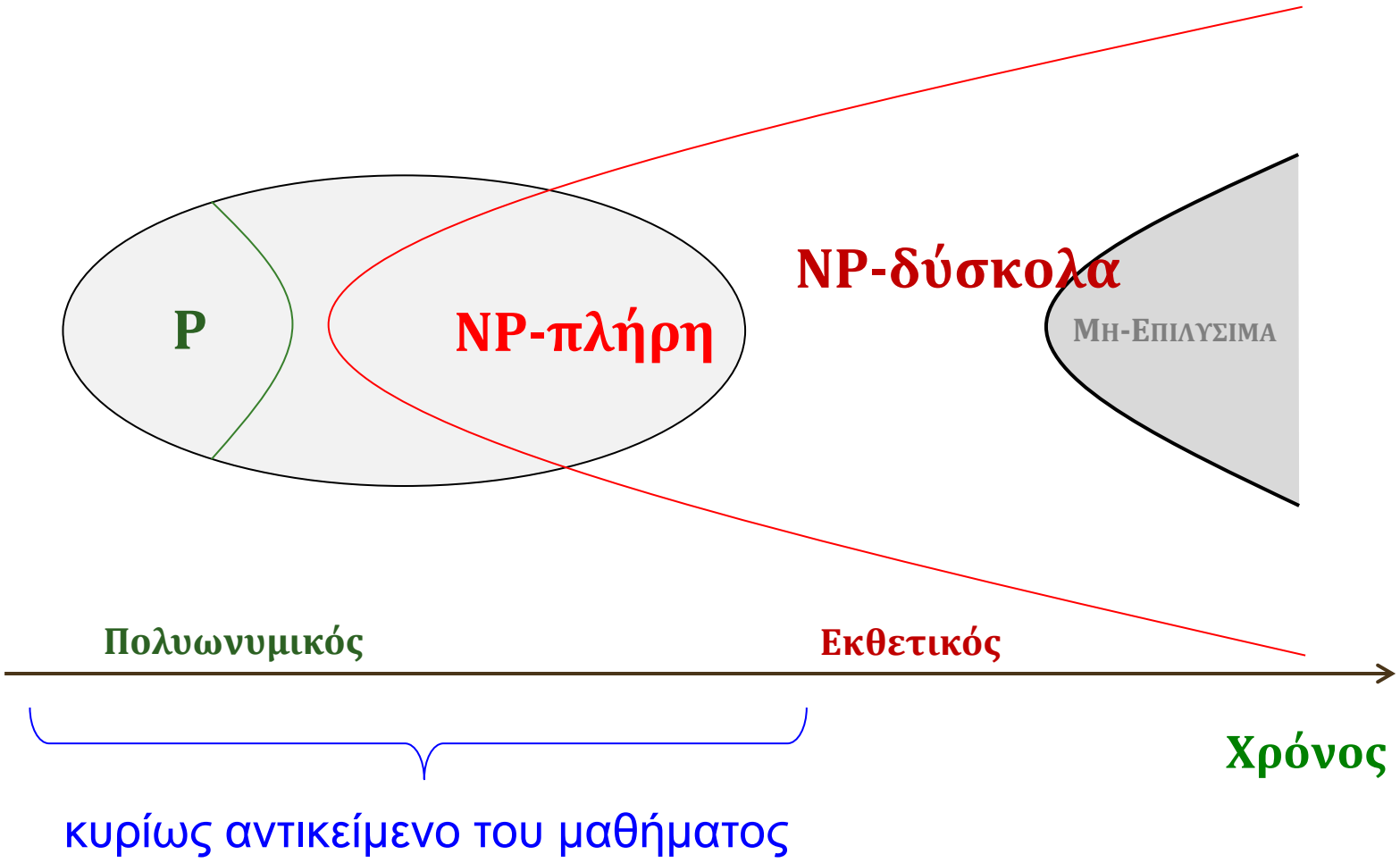
Σημασία **NP**-πληρότητας

"I can't find an efficient algorithm, but neither can all these famous people."



- Αν δεν βρίσκουμε αποδοτικό αλγόριθο για πρόβλημα Π, αλλά δείξουμε ότι είναι **NP**-πλήρες τότε έχουμε δείξει ότι κανείς άλλος στον κόσμο δεν μπορεί!

Κλάσεις πολυπλοκότητας



Ευεπίλυτα και δυσεπίλυτα προβλήματα

- Κλάση **P:** Euler Tour, Reachability, Shortest Paths, Minimum Spanning Tree, Max Flow, Perfect Matching, Linear Programming, …

- **NP**-πλήρη: Satisfiability, Vertex Cover, Clique, Independent Set, Hamilton Cycle, Traveling Salesman Problem, 3-Colorability, Integer Programming, …

Ενδιάμεση πολυπλοκότητα;

- ΠΑΡΑΓΟΝΤΟΠΟΙΗΣΗ
- ΔΙΑΚΡΙΤΟΣ ΛΟΓΑΡΙΘΜΟΣ: λογάριθμος σε αριθμητική modulo
- Είναι στο **NP**, αλλά μάλλον όχι **NP**-πλήρη.
- Κομβική σημασία για σύγχρονη κρυπτογραφία:

  το κρυπτοσύστημα RSA, η ανταλλαγή κλειδιού Diffie-Hellman και πάρα πολλά πρωτόκολλα βασίζονται στη δυσκολία της Παραγοντοποίησης ή/και του Διακριτού Λογάριθμου.

Σημασία Α&Π

- Πολλές επαναστατικές ιδέες και εφαρμογές του καιρού μας στηρίζονται σε αλγόριθμους και πολυπλοκότητα:
  - στο πόσο εύκολα (γρήγορα) μπορούμε να υπολογίσουμε κάποια πράγματα
  - και στο πόσο δύσκολο είναι να υπολογίσουμε κάποια άλλα
- Εφαρμογές: κρυπτογραφία, εκλογές, θεωρία παιγνίων, βιολογία, big data, …

Συμπεράσματα

- Είναι σημαντικό να γνωρίζουμε **τι** μπορούμε να κάνουμε με υπολογιστή, **πώς**, και **πόσο καλά**
- Αυτό θα μελετήσουμε στο μάθημα
- Η εξοικείωση με τις μεθοδολογίες σχεδιασμού αλγορίθμων, και ανάλυσης υπολογιστικής πολυπλοκότητας είναι απαραίτητη για να κατανοούμε τη σύγχρονη τεχνολογία
- …και για να συμμετέχουμε στην ανάπτυξή της
- Τα μαθηματικά είναι πάντα επίκαιρα!