

# Αλγόριθμοι & Πολυπλοκότητα

Διδάσκοντες: **Ε. Ζάχος, Δ. Φωτάκης**

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών

Εθνικό Μετσόβιο Πολυτεχνείο



## Υπολογιστικό Πρόβλημα

- Μετασχηματισμός δεδομένων εισόδου σε δεδομένα εξόδου.
- **Στιγμιότυπο**: μαθηματικό αντικείμενο που ορίζεται από δεδομένα εισόδου.
  - Διατυπώνουμε **ερώτηση** και περιμένουμε **απάντηση**.
  - Άπειρο σύνολο στιγμιότυπων.
- Κατηγορίες Προβλημάτων:
  - **Βελτιστοποίησης**: λύση με βέλτιστη αντικειμενική τιμή.
  - **Απόφασης**: απάντηση **ΝΑΙ** ή **ΌΧΙ**.

## Αντικείμενο

- Ποια υπολογιστικά προβλήματα είναι **εύκολο** και ποια **δύσκολο** να λυθούν από υπολογιστικές μηχανές.
- **Αλγόριθμος**: λεπτομερής περιγραφή **μεθόδου** επίλυσης προβλήματος από υπολογιστική μηχανή.
- Ελάχιστοι υπολογιστικοί πόροι για επίλυση:
  - **Ευεπίλυτα**: spanning tree, shortest paths, max flow, min-cost flow, linear programming, ...
  - **Δυσεπίλυτα**: TSP, knapsack, vertex cover, set cover, scheduling, ...
  - Επίδραση υπολογιστικού μοντέλου.

## Αποδοτική Επίλυση: Κλάση P

- **Μέγεθος στιγμιότυπου  $n$** : αριθμός bits για αναπαράστασή του.
- **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, ...
- **Αξίωμα Cook-Karp**: κλάση **ευεπίλυτων** προβλημάτων ταυτίζεται με **κλάση P**.

# Αλγόριθμοι

- **Τεχνικές** για σχεδιασμό αποδοτικών αλγορίθμων:
  - Διαίρει-και-Βασίλευε
  - Δυναμικός Προγραμματισμός
  - Απληστία
- Εφαρμογή στην **επίλυση βασικών** προβλημάτων:
  - Ταξινόμηση, Min Spanning Tree, Shortest Paths, Max Flow, ...

# «Δύσκολα» Προβλήματα

- Τι κάνουμε όταν ένα **πρόβλημα** φαίνεται «**δύσκολο**»;
  - «Δύσκολο»: μετά από μεγάλη προσπάθεια, δεν βρίσκουμε αποδοτικό αλγόριθμο (πολυωνυμικού χρόνου).
- Πάμε στο αφεντικό και λέμε:
  - Δεν **μπορώ** να βρω αποδοτικό αλγόριθμο. **Απόλυση!**
  - Δεν **υπάρχει** αποδοτικός αλγόριθμος. **Καλό αλλά δύσκολο!**
  - **Κανένας** δεν μπορεί να βρει αποδοτικό αλγόριθμο (και όλοι πιστεύουν ότι δεν υπάρχει).
- Θεωρία **NP-πληρότητας**.
  - **NP-πλήρη**: κλάση εξαιρετικά **σημαντικών προβλημάτων** που είτε όλα επιλύονται σε πολυωνυμικό χρόνο είτε κανένα.

# Η Κλάση NP

- ...περιλαμβάνει προβλήματα απόφασης:
  - Για κάθε **ΝΑΙ-στιγμιότυπο**, υπάρχει «πιστοποιητικό» **εύκολο να ελεγχθεί** (ελέγχεται σε πολυωνυμικό χρόνο).
    - Συνοπτικό πιστοποιητικό (succinct certificate).
  - Πιστοποιητικό μπορεί να είναι **δύσκολο να υπολογισθεί**.  
Αν δοθεί, **ελέγχεται εύκολα!**
  - Δεν απαιτείται συνοπτικό πιστοποιητικό για **ΌΧΙ-στιγμιότυπα**.
- Για **σημαντικότερα προβλήματα** βελτιστοποίησης, αντίστοιχα προβλήματα απόφασης **ανήκουν στο NP**.

# Υπολογιστική Πολυπλοκότητα

- Μηχανές Turing, μη υπολογισιμότητα.
- Κλάσεις υπολογιστικής πολυπλοκότητας, αναγωγή και πληρότητα.
- NP-πληρότητα και συνέπειες.
  - (Αλγοριθμική) αντιμετώπιση NP-δύσκολων προβλημάτων
  - Προσεγγισιμότητα και μη-προσεγγισιμότητα.
- Κλάσεις πολυπλοκότητας για πιθανοτικό υπολογισμό.
- Κρυπτογραφία.
- Χωρική πολυπλοκότητα.

## Βιβλιογραφία

---

- Cormen, Leiserson, Rivest, Stein. **Introduction to Algorithms**. MIT Press, 2001.
- Kleinberg, Tardos. **Algorithm Design**. Add.-Wesl., 2006.
- Dasgupta, Papadimitriou, Vazirani. **Algorithms**. McGraw-Hill, 2008.
- Brassard, Bratley. **Algorithmics: Theory and Practice**. Prentice-Hall, 1988.
- Papadimitriou. **Computational Complexity**. Add.-Wesl., 1994.
- Arora, Barak. **Computational Complexity: A Modern Approach**. Cambridge, 2009.
- Goldreich. **Computational Complexity: A Conceptual Perspective**. Cambridge, 2009.

## Πληροφορίες

---

- **Δευτέρα:** (15:00-) **16:00-18:00**, Αμφ. 5, Νέο Κτήριο ΣΗΜΜΥ.
- **Πέμπτη:** **16:00-18:00** (-19:00), Αμφ. Ηλεκτρολ., Παλ. Κτήριο ΣΗΜΜΥ.
- <http://www.corelab.ece.ntua.gr/courses/grad-algo/>