[](http://www.teilar.gr/)

**Αλγόριθμοι και Στοιχεία Πολυπλοκότητας.**

**Ενότητα 12:** Οι Κλάσεις P και NP.

Διδάσκων: Ηλίας Κ Σάββας, Αναπληρωτής Καθηγητής.

Τμήμα Μηχανικών Πληροφορικής, Τεχνολογικής Εκπαίδευσης.

**Άδειες χρήσης.**

* Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons (C C). **Αναφορά δημιουργού (B Y), Μη εμπορική χρήση (N C), Μη τροποποίηση (N D), 3.0, Μη εισαγόμενο.**
* Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

[](http://creativecommons.org/licenses/by-nc-nd/3.0/deed.el)

**Χρηματοδότηση.**

* Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

[](http://www.edulll.gr/)

# Σκοποί ενότητας.

Ο αναγνώστης να μπορεί να:

αντιληφθεί τις έννοιες των κλάσεων προβλημάτων P και NP.

# ΠΕΡΙΕΧΌΜΕΝΑ ΕΝΌΤΗΤΑΣ.

[Σκοποί ενότητας. 2](#_Toc368164015)

[ΠΕΡΙΕΧΌΜΕΝΑ ΕΝΌΤΗΤΑΣ. 3](#_Toc368164016)

[ΑΛΓΌΡΙΘΜΟΙ. 3](#_Toc368164017)

[ΣΧΉΜΑΤΑ. 3](#_Toc368164018)

[12. ΟΙ ΚΛΆΣΕΙΣ P ΚΑΙ NP. 4](#_Toc368164019)

[12.1 Αιτιοκρατικοί και μη αιτιοκρατικοί αλγόριθμοι. 4](#_Toc368164020)

[12.2 Η κλάση προβλημάτων P. 5](#_Toc368164021)

[12.3 Η κλάση προβλημάτων NP. 5](#_Toc368164022)

[12.4 Η κλάση των NP-hard και NP-complete προβλημάτων. 6](#_Toc368164023)

# ΑΛΓΌΡΙΘΜΟΙ.

|  |  |
| --- | --- |
| ***Αλγόριθμος 1****: Μη αιτιοκρατικός αλγόριθμος αναζήτησης.* | 4 |
| ***Αλγόριθμος 2****: Μη αιτιοκρατικός αλγόριθμος ταξινόμησης.* | 5 |

# ΣΧΉΜΑΤΑ.

|  |  |
| --- | --- |
| ***Σχήμα 12.1****: Σχέση κλάσεων P και NP με την υπόθεση.* | 6 |
| ***Σχήμα 12.2****: Σχέσεις των κλάσεων P, NP, NP-hard, και NP-complete.* | 7 |

# 12. ΟΙ ΚΛΆΣΕΙΣ *P* ΚΑΙ *NP.*

Όπως έχει ήδη ορισθεί, η πολυπλοκότητα ενός αλγόριθμου είναι το σύνολο των βημάτων, (στοιχειωδών πράξεων, όπως πρόσθεση, σύγκριση, και τα λοιπά), που χρειάζεται για να επιλύσει κάποιο πρόβλημα. Επίσης έχει ήδη συζητηθεί, ότι αυτή η πολυπλοκότητα εξαρτάται από το πλήθος των δεδομένων εισόδου. Δηλαδή, μπορεί να αποδοθεί από μία συνάρτηση , όπου για κάθε συγκεκριμένο πλήθος δεδομένων εισόδου *n,* επιστρέφει το πλήθος των στοιχειωδών πράξεων , που απαιτούνται για την επίλυση του προβλήματος.

## 12.1 Αιτιοκρατικοί και μη αιτιοκρατικοί αλγόριθμοι.

Όλοι οι αλγόριθμοι που έχουν εξετασθεί μέχρι τώρα, έχουν την ιδιότητα του να είναι καλά, και μάλιστα μοναδικά ορισμένοι. Δηλαδή, να προσδιορίζουν ακριβώς το κάθε βήμα τους, και να μην υπάρχει η παραμικρή αμφιβολία σε όλα τα στάδια της εκτέλεσής τους. Αυτοί οι αλγόριθμοι ονομάζονται αιτιοκρατικοί ή ντετερμινιστικοί (*deterministic*). Και βέβαια συμφωνούν απόλυτα, με το πώς ένας υπολογιστής αντιλαμβάνεται ένα πρόγραμμα. Από την θεωρητική σκοπιά όμως, είναι δυνατόν να απαλειφθεί αυτός ο παράγοντας. Δηλαδή, να είναι δυνατή η μη καλά ορισμένη λειτουργία κάποιου σημείου του αλγόριθμου, και μάλιστα να επιτρέπεται να επιλέξει από ένα σύνολο πιθανών αποφάσεων, αντί της μίας και μοναδικής. Αυτή η εξέλιξη οδηγεί στους λεγόμενους μη-αιτιοκρατικούς αλγόριθμους. Η γενική λειτουργία τους συνοψίζεται ως εξής:

1. Επιλογή(*S*): «τυχαία» επιλογή ενός στοιχείου του συνόλου *S.*
2. Αποτυχία(): ένδειξη μη επιτυχούς κατάληξης.
3. Επιτυχία(): ένδειξη επιτυχούς κατάληξης.

*Παράδειγμα 1*: Αναζήτηση στοιχείου *x* σε πίνακα *A*N. Πέραν των κλασσικών προσδιοριστικών αλγόριθμων αναζήτησης σε πίνακα, ένας μη αιτιοκρατικός αλγόριθμος, θα μπορούσε να περιγράψει το πρόβλημα όπως φαίνεται στον *Αλγόριθμο 1*. Η πολυπλοκότητα του αλγόριθμου είναι της τάξης .

***Αλγόριθμος 1****: Μη αιτιοκρατικός αλγόριθμος αναζήτησης.*

1: j 🡨 Επιλογή(1,2,…,*N*).

2: Εάν (*x* = *A*j), Τότε,

3: Επέστρεψε j,

4: Επιτυχία().

5: Αλλιώς,

6: Αποτυχία(),

7: Τέλος Εάν.

8: Τέλος Αλγόριθμου «Μη αιτιοκρατικός αλγόριθμος αναζήτησης».

*Παράδειγμα 2*: Ταξινόμηση πίνακα AN. Στον *Αλγόριθμο 2*, περιγράφεται ένας μη αιτιοκρατικός αλγόριθμος ταξινόμησης πίνακα. Ο αλγόριθμος ταξινομεί τον πίνακα σε φθίνουσα διάταξη, και χρησιμοποιεί έναν βοηθητικό πίνακα *B*N. Πρώτα, αρχικοποιεί τον πίνακα B και στην συνέχεια τοποθετεί στοιχεία του πίνακα *A* στον *B*. Μετά το τέλος αυτής της διαδικασίας, ο πίνακας B στην ουσία αποτελεί μία αναδιάταξη των στοιχείων του A, και για αυτό ο αλγόριθμος ελέγχει να δει εάν ο B είναι ταξινομημένος ή όχι, και επιστρέφει αντίστοιχα επιτυχία ή αποτυχία. Η πολυπλοκότητα του αλγόριθμου είναι της τάξης .

***Αλγόριθμος 2****: Μη αιτιοκρατικός αλγόριθμος ταξινόμησης.*

1: Για i 🡨1 Μέχρι *Ν,*

2: Bi 🡨 0, /\* Αρχικοποίηση B \*/,

3: Τέλος Για (i).

4: Για i 🡨 1 Μέχρι *N,*

5: j 🡨 Επιλογή(1,2,…,*N*),

6: Εάν (*B*j ≠ 0). Τότε,

7: Bj 🡨 Ai,

8: Τέλος Εάν.

9: Τέλος Για (i).

10: Για i 🡨 1 Μέχρι *N,*

11: Εάν (*Β*i > *B*i+1). Τότε,

12: Επέστρεψε Αποτυχία(),

13: Τέλος Εάν.

14: Τέλος Για (i).

15: Επέστρεψε Επιτυχία().

16: Τέλος αλγόριθμου «Μη αιτιοκρατικός ταξινόμησης».

## 12.2 Η κλάση προβλημάτων *P.*

Αν ένα πρόβλημα αποσκοπεί στη λήψη μιας δυαδικής απόφασης, (ναι ⁄ όχι), τότε λέγεται πρόβλημα απόφασης. Από τα προβλήματα απόφασης όσα επιδέχονται λύση από πολυωνυμικούς αλγόριθμους, αποτελούν την κλάση (οικογένεια) *P*. Δηλαδή, ένας αλγόριθμος θεωρείται ότι είναι υπολογιστικής πολυπλοκότητας της τάξης *P*, εάν υπάρχει ένα πολυώνυμο *p*() τέτοιο ώστε, η πολυπλοκότητα του αλγόριθμου να μπορεί να εκφρασθεί σαν *,* για οποιοδήποτε πλήθος δεδομένων *N*.

## 12.3 Η κλάση προβλημάτων *NP.*

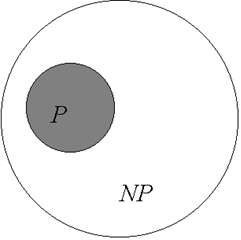
Έστω ένα πρόβλημα απόφασης μεγέθους *n* της μορφής:

«Τα δεδομένα , ικανοποιούν την συνθήκη *C*».

Δηλαδή για παράδειγμα, ας θεωρηθεί το ακόλουθο πρόβλημα: Δοθέντος ενός φυσικού αριθμού *m*, με δυαδική αναπαράσταση , όπου , να ελεγχθεί εάν αυτός ο αριθμός είναι σύνθετος, δηλαδή εάν έχει ακέραιους διαιρέτες.

Έστω τώρα ότι η εν λόγω συνθήκη, ικανοποιείται αν βρεθεί έστω και ένας διαιρέτης, (εκτός φυσικά της μονάδας και του εαυτού του).

Ας υποτεθεί τώρα ότι ένας «από μηχανής Θεός» προτείνει έναν αριθμό *b* κάποιου μεγέθους, ο οποίος ισχυρίζεται ότι διαιρεί τον *a*. Εάν ο «από μηχανής Θεός» είναι αξιόπιστος, το πρόβλημα παίρνει την μορφή του ελέγχου εάν ο *b* διαιρεί τον *a*. Εάν το νέο αυτό πρόβλημα ανήκει στην κλάση *P*, τότε ορίζεται πως το αρχικό πρόβλημα ανήκει στην κλάση *NP*. Προφανώς ισχύει . Ωστόσο παραμένει αναπόδεικτη η εικασία . Αυτό δείχνει την άγνοια του ότι ενώ είναι γνωστά προβλήματα της κλάσης *NP,* δεν μπορεί να αποδειχθεί, ότι δεν μπορεί να βρεθούν κάποιοι καλύτεροι αλγόριθμοι, για αυτά τα προβλήματα, ώστε να ανήκουν στην κλάση *P* (*Σχήμα 12.1*).



***Σχήμα 12.1****: Σχέση κλάσεων P και NP με την υπόθεση .*

***Ορισμός 12.1***: Το σύνολο των προσδιοριστικών αλγόριθμων απόφασης που επιλύονται σε πολυωνυμικό χρόνο, ανήκουν στην κλάση *P*.

***Ορισμός 12.2***: Το σύνολο των αλγόριθμων απόφασης που επιλύονται σε πολυωνυμικό χρόνο, από μη προσδιοριστικούς αλγόριθμους, ανήκουν στην κλάση *ΝP*.

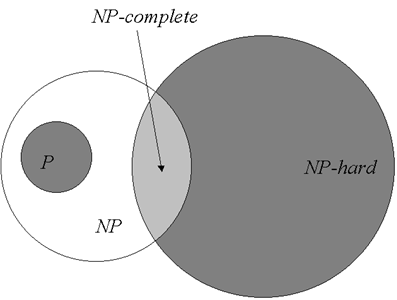
## 12.4 Η κλάση των *NP*-*hard* και *NP*-*complete* προβλημάτων.

Σαν συνέπεια της προαναφερθείσης άγνοιας, ορίζεται μία κλάση προβλημάτων *NP*-complete (*NPC*). Τα προβλήματα που ανήκουν σε αυτή την κλάση έχουν τις εξής δύο ιδιότητες:

Ανήκουν στην κλάση *NP*.

Αν έστω και για ένα από αυτά αποδειχθεί (στο μέλλον), ότι ανήκει στην κλάση *P,* τότε αυτόματα έχει αποδειχθεί ότι , δηλαδή όλα τα *NP*-complete προβλήματα, θα ήταν επιλύσιμα σε πολυωνυμικό χρόνο.

Από τα παραπάνω είναι σαφές ότι από τα *NP* προβλήματα, τα *P* είναι τα εύκολα και τα *NP*-complete τα δύσκολα, ως προς την υπολογιστική τους πολυπλοκότητα. Από την άλλη μεριά όμως μπορεί να αποδειχθεί ότι όλα τα *NP* προβλήματα, συμπεριλαμβανομένων και των *NP*-complete, επιδέχονται αλγόριθμους με πολυπλοκότητα . Είναι δηλαδή λιγότερο ή το πολύ ισοδύναμοι με αλγόριθμους εκθετικής πολυπλοκότητας.



***Σχήμα 12.2****: Σχέσεις των κλάσεων P, NP, NP-hard, και NP-complete.*

***Ορισμός 12.3***: Έστω *L*1 και *L*2 είναι δύο προβλήματα απόφασης. Ορίζεται ότι το πρόβλημα *L*1, περιορίζει το πρόβλημα *L*2 (*L*1  *L*2), εάν και μόνον εάν υπάρχει ένας τρόπος, ώστε να επιλυθεί το *L*1 από προσδιοριστικό αλγόριθμο σε πολυωνυμικό χρόνο, χρησιμοποιώντας έναν προσδιοριστικό αλγόριθμο που επιλύει το *L*2.

***Ορισμός 12.4***: Ένα πρόβλημα *L,* θεωρείται ότι ανήκει στην κλάση *NP-hard,* εάν η ικανοποίηση του προβλήματος περιορίζει το L (ικανοποίηση *L*).

***Ορισμός 12.5***: Ένα πρόβλημα *L,* θεωρείται ότι ανήκει στην κλάση *NP-complete,* εάν και μόνο εάν το *L* ανήκει στην κλάση *NP-hard,* και ταυτόχρονα στην κλάση *NP*.

Οι σχέσεις που υπάρχουν μεταξύ των διαφόρων κλάσεων προβλημάτων φαίνονται στο *Σχήμα 12.2*.

Τέλος δωδέκατης ενότητας.

[](http://creativecommons.org/licenses/by-nc-nd/3.0/deed.el)[](http://www.edulll.gr/)