



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Προγραμματιστικές Τεχνικές

Βασίλειος Βεσκούκης

Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Ρωμύλος Κορακίτης

Αστροφυσικός
Αναπλ. Καθηγητής ΕΜΠ
romylos@survey.ntua.gr

Αλγόριθμοι ταξινόμησης σε πίνακες

Ταξινόμηση (sort)

Μια από τις πιο συνηθισμένες και σημαντικές διαδικασίες επεξεργασίας δεδομένων είναι η ταξινόμηση μιας σειράς (λίστας) ομοειδών δεδομένων, αριθμητικών ή αλφαριθμητικών.

Ταξινόμηση (sorting) είναι η διαδικασία μετατροπής μιας λίστας στοιχείων (ενός πίνακα) σε μια *ισοδύναμη* λίστα όπου τα στοιχεία είναι τοποθετημένα κατά σειρά, αύξουσα ή φθίνουσα, σύμφωνα με την τιμή τους.

Δηλαδή, αν η ταξινομημένη λίστα είναι καταχωρημένη στον πίνακα $A[N]$ ισχύει:

$$A[0] \leq A[1] \leq A[2] \leq \dots \leq A[N - 2] \leq A[N - 1]$$

Πολλές μέθοδοι (αλγόριθμοι) έχουν προταθεί για την ταξινόμηση των στοιχείων ενός πίνακα. Βέβαια, οι απλούστεροι αλγόριθμοι είναι λιγότερο αποδοτικοί από τους πιο πολύπλοκους, κάτι που έχει σημασία όταν το μέγεθος του πίνακα είναι πολύ μεγάλο.

Ταξινόμηση (sort)

Παράδειγμα 1:

Αρχική λίστα: 43 , 22 , 17 , 36 , 16 , 28

Ταξινομημένη λίστα: 16 , 17 , 22 , 28 , 36 , 43 (αύξουσα ταξινόμηση)

43 , 36 , 28 , 22 , 17 , 16 (φθίνουσα ταξινόμηση)

Παράδειγμα 2:

Αρχική λίστα: Q , W , E , R , T , Y , P

Ταξινομημένη λίστα: E , P , Q , R , T , W , Y (αύξουσα ταξινόμηση)

Στην περίπτωση ταξινόμησης αλφαριθμητικών στοιχείων, ως τιμή του κάθε στοιχείου λαμβάνεται ο αριθμός της κωδικοποίησης του αλφαριθμητικού (συνήθως ASCII).

Η ταξινόμηση μιας λίστας έχει μεγάλη σημασία γιατί διευκολύνει σημαντικά την αναζήτηση στοιχείων μέσα στην λίστα.

Ταξινόμηση (sort)

Όλοι οι αλγόριθμοι ταξινόμησης βασίζονται σε **σύγκριση** των στοιχείων και σε **αντιμετάθεση** των στοιχείων, όπου χρειάζεται.

Για την ευκολότερη υλοποίηση των αλγορίθμων αυτών είναι πολύ χρήσιμη μια συνάρτηση αντιμετάθεσης στοιχείων, της μορφής:

Συνάρτηση **Αντιμετάθεση** (Πίνακας $A[]$, δείκτης K , δείκτης M)

$$\text{Temp} = A[K]$$
$$A[K] = A[M]$$
$$A[M] = \text{Temp}$$

Δύο χαρακτηριστικοί αλγόριθμοι ταξινόμησης είναι:

- Αλγόριθμος **επιλογής** (selection sort)
- Αλγόριθμος **φουσαλίδας** (bubble sort)

Ταξινόμηση με επιλογή (Selection sort)

Ο πίνακας $A[N]$ περιέχει αταξινομήτα δεδομένα και επιζητούμε **αύξουσα** ταξινόμηση.

Η λειτουργία του αλγορίθμου αυτού είναι: διαδοχική επιλογή του μεγαλύτερου στοιχείου και τοποθέτησή του στο άνω άκρο του πίνακα, το οποίο μειώνεται κατά ένα σε κάθε επανάληψη. Δηλαδή:

1^η επανάληψη: Αναζήτηση του μέγιστου των N στοιχείων και αντιμετάθεσή του με το στοιχείο $A[N - 1]$

2^η επανάληψη: Αναζήτηση του μέγιστου των υπολοίπων $(N - 1)$ στοιχείων και αντιμετάθεσή του με το στοιχείο $A[N - 2]$

3^η επανάληψη: Αναζήτηση του μέγιστου των υπολοίπων $(N - 2)$ στοιχείων και αντιμετάθεσή του με το στοιχείο $A[N - 3]$

....

$(N - 1)$ ^η επανάληψη: Αναζήτηση του μέγιστου των τελευταίων 2 στοιχείων και αντιμετάθεσή του με το στοιχείο $A[1]$

Ταξινόμηση με επιλογή (Selection sort)

ΨΕΥΔΟΚΩΔΙΚΑΣ

ΑΡΧΗ

B1 Για $K = N - 1$ ως 1 , βήμα -1 , επανάλαβε:

$M = 0$

B2 Για $I = 1$ ως K , βήμα $+1$, επανάλαβε:

B3 Αν $A[I] > A[M]$, τότε $M = I$

Τέλος επανάληψης (I)

B4 Αντιμετάθεσε τις τιμές των $A[K]$ και $A[M]$

Τέλος επανάληψης (K)

ΤΕΛΟΣ

Ταξινόμηση με επιλογή (Selection sort)

ΠΑΡΑΔΕΙΓΜΑ

1^η επανάληψη (Αρχική λίστα): 43 , 22 , 17 , 36 , 16 , 28

Μέγιστο : 43 → τροποποιημένη λίστα: 28, 22, 17, 36, 16, 43

2^η επανάληψη : 28, 22, 17, 36, 16, 43

Μέγιστο : 36 → τροποποιημένη λίστα: 28, 22, 17, 16, 36, 43

3^η επανάληψη : 28, 22, 17, 16, 36, 43

Μέγιστο : 28 → τροποποιημένη λίστα: 16, 22, 17, 28, 36, 43

4^η επανάληψη : 16, 22, 17, 28, 36, 43

Μέγιστο : 22 → τροποποιημένη λίστα: 16, 17, 22, 28, 36, 43

5^η επανάληψη : 16, 17, 22, 28, 36, 43

Μέγιστο : 17 → ταξινομημένη λίστα: 16, 17, 22, 28, 36, 43

Ταξινόμηση φουσαλίδας (Bubble sort)

Ο πίνακας $A[N]$ περιέχει αταξινομήτα δεδομένα και επιζητούμε **αύξουσα** ταξινόμηση.

Η λειτουργία του αλγορίθμου αυτού είναι: επαναληπτικά περάσματα των στοιχείων του πίνακα έτσι ώστε, σε κάθε πέρασμα, το μεγαλύτερο από τα στοιχεία να 'σπρώχνεται' (με διαδοχικές συγκρίσεις και αντιμεταθέσεις) προς το άνω άκρο του πίνακα, το οποίο μειώνεται κατά ένα σε κάθε πέρασμα. Η επανάληψη τερματίζεται όταν εξαντληθούν όλα τα περάσματα ή όταν σε κάποιο πέρασμα δεν χρειαστεί καμιά αντιμετάθεση στοιχείων.

Ταξινόμηση φουσαλίδας (Bubble sort)

ΨΕΥΔΟΚΩΔΙΚΑΣ

ΑΡΧΗ

B1 Πέρασμα = 1

B2 Αντιμετάθεση = ΑΛΗΘΗΣ

B3 **Εφ όσον** (Πέρασμα < N ΚΑΙ Αντιμετάθεση = ΑΛΗΘΗΣ) , **εκτέλεσε:**
Αντιμετάθεση = ΨΕΥΔΗΣ

B4 Για $I = 0$ ως $(N - \text{Πέρασμα})$, βήμα +1 , **επανάλαβε:**

B5 Αν $A[I] > A[I + 1]$, τότε : **Αντιμετάθεσε** $A[I]$ και $A[I + 1]$
Αντιμετάθεση = ΑΛΗΘΗΣ

Τέλος επανάληψης (I)

B6 Πέρασμα = Πέρασμα + 1

B7 *Επιστροφή στο B3*

ΤΕΛΟΣ

Ταξινόμηση φουσαλίδας (Bubble sort)

ΠΑΡΑΔΕΙΓΜΑ

(Αρχική λίστα): 43 , 22 , 17 , 36 , 16 , 28

1^ο Πέρασμα: 22 , 43 , 17 , 36 , 16 , 28

22 , 17 , 43 , 36 , 16 , 28

22 , 17 , 36 , 43 , 16 , 28

22 , 17 , 36 , 16 , 43 , 28

22 , 17 , 36 , 16 , 28 , 43

2^ο Πέρασμα: 17 , 22 , 36 , 16 , 28 , 43

17 , 22 , 36 , 16 , 28 , 43

17 , 22 , 16 , 36 , 28 , 43

17 , 22 , 16 , 28 , 36 , 43

3^ο Πέρασμα: 17 , 22 , 16 , 28 , 36 , 43

Κ.Ο.Κ.

ταξινομημένη λίστα: 16, 17, 22, 28, 36, 43

Σύγκριση των αλγορίθμων ταξινόμησης

Ο αλγόριθμος επιλογής εκτελεί $1 + 2 + 3 + \dots + (N - 1)$ συγκρίσεις και, κατά μέσο όρο, $N/2$ αντιμεταθέσεις στοιχείων (που χρειάζονται 3 αναθέσεις τιμής η κάθε μία). Επειδή ισχύει :

$1 + 2 + 3 + \dots + N = N(N - 1) / 2$, η μέση επίδοση του αλγορίθμου είναι ανάλογη του N^2 .

Ο αλγόριθμος φουσαλίδας εκτελεί, κατά μέσο όρο, $(N - 1)/2$ περάσματα, $N/2$ συγκρίσεις και $N/4$ αντιμεταθέσεις στο καθένα, επομένως η επίδοση και αυτού του αλγορίθμου είναι ανάλογη του N^2 .

Χρησιμοποιώντας σύνθετους αλγορίθμους, που χρησιμοποιούν **αναδρομή** (π.χ. QuickSort) μπορούμε να επιτύχουμε μέση επίδοση ανάλογη του $N \log N$, που αποτελεί σημαντική βελτίωση για μεγάλες τιμές του N .

Παράδειγμα: Ταξινόμηση πίνακα με 10^6 στοιχεία μπορεί να απαιτεί 10^{12} συγκρίσεις με αλγόριθμο επιλογής και μόνο 10^7 συγκρίσεις με αλγόριθμο QuickSort.

Παράδειγμα - Ταξινόμηση πίνακα με bubblesort

Πρόβλημα:

Οι ορθογώνιες συντεταγμένες (x,y) ενός αριθμού N σημείων ($N < 100$) είναι καταχωρημένες στο αρχείο `points.txt`. Να κατασκευαστεί ένα πρόγραμμα που θα εκτελεί τα ακόλουθα:

1. Θα διαβάζει τις συντεταγμένες από το αρχείο και θα τις εκχωρεί σε αντίστοιχους πίνακες.
2. Θα υπολογίζει τα μήκη των ευθυγράμμων τμημάτων μεταξύ διαδοχικών σημείων.
3. Θα ταξινομεί, με αύξουσα σειρά, τον πίνακα που θα περιέχει τα μήκη και θα τον γράφει στο αρχείο `sortlen.txt`.
4. Θα υπολογίζει την μέση τιμή, την τυπική απόκλιση και την διάμεση τιμή των μηκών και θα γράφει τα μεγέθη αυτά στο αρχείο `sortlen.txt`.

Παράδειγμα - Ταξινόμηση πίνακα με bubblesort

Δείγμα εγγραφών του αρχείου points.txt:

53.19 15.55

52.93 19.29

54.25 22.57

42.64 24.10

.....

Μορφοποίηση εγγραφών του αρχείου sortlen.txt:

Sorted lengths

2.034

2.256

.....

12.781

13.294

Mean value = 6.256

Standard Deviation = 2.873

Median value = 7.012