



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

Αλγόριθμοι

Στοιχειώδεις μέθοδοι ταξινόμησης

Τσούρος Δήμος

- **Εισαγωγή**
- **Αλγόριθμος ταξινόμησης με εισαγωγή**
- **Αλγόριθμος ταξινόμησης με επιλογή**
- **Αλγόριθμος ταξινόμησης φουσαλίδας**
- **Σύγκριση στοιχειωδών αλγόριθμων ταξινόμησης**

- **Πρόβλημα:** Δίνεται πίνακας $T = [t_1 \ t_2 \ \cdots \ t_n]$ και ζητείται να ταξινομηθεί (κατά αύξουσα τάξη) έτσι ώστε

$$t_1 \leq t_2 \leq \cdots \leq t_n$$

- Πιθανότατα το **περισσότερο χρησιμοποιούμενο** πρόβλημα της πληροφορικής
- Κριτήρια **αποδοτικότητας** μιας μεθόδου ταξινόμησης
 - ο αριθμός των συγκρίσεων
 - ο αριθμός των μετακινήσεων

- Ταξινόμηση ακεραίων

12	4	24	1	43	32	11	31	42
0	1	2	3	4	5	6	7	8

1	4	11	12	24	31	32	42	43
0	1	2	3	4	5	6	7	8

ΑΛΓΟΡΙΘΜΟΙ ΤΑΞΙΝΟΜΗΣΗΣ

Αλγόριθμος	Καλύτερη περίπτωση	Αναμενόμενη περίπτωση	Χειρότερη περίπτωση	Χώρος	Ευστάθεια
Bubble sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Σταθερός	Ναι
Modified bubble sort	$O(n^2)$	$O(n)$	$O(n^2)$	Σταθερός	Ναι
Insertion sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Σταθερός	Ναι
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Σταθερός	Ναι
Heap sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	Σταθερός	Όχι
Merge sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	Μη σταθερός	Ναι
Quick sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	Σταθερός	Ναι

Ποιος είναι ο καλύτερος αλγόριθμος ταξινόμησης;

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (1)

- **Ταξινόμηση με εισαγωγή (Insertion sort)**
- Ένας από τους απλούστερους αλγόριθμους ταξινόμησης με την εξής λειτουργία
 - ξεκινάμε από τη δεύτερη θέση και αν είναι μικρότερο από το πρώτο, τότε τα αντιμεταθέτουμε, αλλιώς δεν κάνουμε τίποτα
 - έπειτα ξεκινάμε από την τρίτη θέση και το συγκρίνουμε με το δεύτερο στοιχείο και αν είναι μικρότερο, τότε το αντιμεταθέτουμε και συνεχίζουμε τη σύγκριση με το πρώτο στοιχείο
 - και συνεχίζει με αυτόν τον τρόπο μέχρι να ταξινομηθεί ολόκληρος ο πίνακας

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (1)

- Ταξινόμηση με εισαγωγή (Insertion sort)

Αλγόριθμος ταξινόμησης με εισαγωγή

1. Είσοδος: πίνακας A με n στοιχεία
2. Για i από 2 μέχρι n :
 1. $key \leftarrow A[i]$
 2. $j \leftarrow i - 1$
 3. Όσο $j > 0$ και $A[j] > key$:
 1. $A[j+1] \leftarrow A[j]$
 2. $j \leftarrow j - 1$
 4. Τέλος_Όσο
 5. $A[j+1] \leftarrow key$
3. Τέλος_Για
4. Έξοδος: ταξινομημένος πίνακας A

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (2)

Εφαρμογή. $T = [15 \ 10 \ 7 \ 2 \ 5 \ 4]$

$i = 2$	$T = [15 \ \underline{10} \ 7 \ 2 \ 5 \ 4]$
$i = 3$	$T = [10 \ 15 \ \underline{7} \ 2 \ 5 \ 4]$
$i = 4$	$T = [7 \ 10 \ 15 \ \underline{2} \ 5 \ 4]$
$i = 5$	$T = [2 \ 7 \ 10 \ 15 \ \underline{5} \ 4]$
$i = 6$	$T = [2 \ 5 \ 7 \ 10 \ 15 \ \underline{4}]$
	$T = [2 \ 4 \ 5 \ 7 \ 10 \ 15]$

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (4)

- **Πλεονεκτήματα**
 - Αποδοτική όταν το αρχείο είναι ήδη ταξινομημένο
 - Αποδοτική όταν το αρχείο είναι μερικώς ταξινομημένο
 - Είναι ευσταθής μέθοδος

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΙΣΑΓΩΓΗ (5)

- **Μειονεκτήματα**
 - Ο χρόνος εκτέλεσης εξαρτάται από το συνολικό πλήθος των αντιστροφών που υπάρχουν στο αρχείο
 - Μη αποδοτική μέθοδος ταξινόμησης όταν τα στοιχεία είναι πολύ μεγαλύτερα από τα κλειδιά
 - Πολυπλοκότητα: $O(n^2)$

- <https://yongdanielliang.github.io/animation/web/InsertionSortNew.html>

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (1)

- **Ταξινόμηση με επιλογή (Selection sort)**
- Ένας από τους απλούστερους αλγόριθμους ταξινόμησης με την εξής λειτουργία
 - πρώτα βρίσκει το μικρότερο στοιχείο του πίνακα και το αντιμεταθέτει με το στοιχείο που βρίσκεται στην πρώτη θέση του πίνακα
 - έπειτα βρίσκει το δεύτερο μικρότερο στοιχείο και το αντιμεταθέτει με το στοιχείο που βρίσκεται στη δεύτερη θέση
 - και συνεχίζει με αυτόν τον τρόπο μέχρι να ταξινομηθεί ολόκληρος ο πίνακας

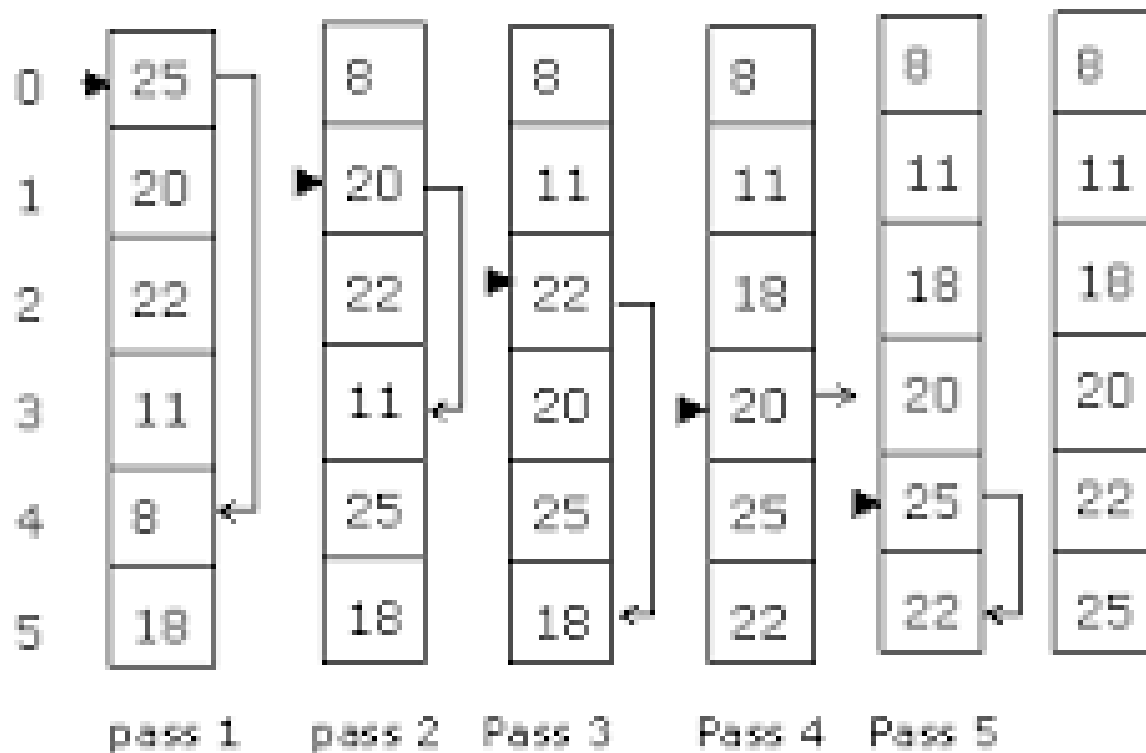
ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (1)

- Ταξινόμηση με επιλογή (Selection sort)

Αλγόριθμος ταξινόμησης με επιλογή

1. Είσοδος: πίνακας A με n στοιχεία
2. Για i από 1 μέχρι n-1
 1. $\text{min} \leftarrow i$
 2. Για j από i+1 μέχρι n
 1. Αν $A[j] < A[\text{min}]$ τότε
 1. $\text{min} \leftarrow j$
 2. Τέλος_Αν
 3. Τέλος_Για
 4. Αν $\text{min} \neq i$ τότε
 1. $\text{Temp} \leftarrow A[i]$
 2. $A[i] \leftarrow A[\text{min}]$
 3. $A[\text{min}] \leftarrow \text{temp}$
 5. Τέλος_Αν
3. Τέλος_Για
4. Έξοδος: ταξινομημένος πίνακας A

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (2)



ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (3)

Παράδειγμα. $T = [10\ 15\ 7\ 2\ 5\ 4]$

εναλλαγή Διάνυσμα T

10 15 7 2 5 4]

$T(1) \leftrightarrow T(4)$ [2 15 7 10 5 4]

$T(2) \leftrightarrow T(6)$ [2 4 7 10 5 15]

$T(3) \leftrightarrow T(5)$ [2 4 5 10 7 15]

$T(4) \leftrightarrow T(5)$ [2 4 5 7 10 15]

$T(5) \leftrightarrow T(5)$ [2 4 5 7 10 15]

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (5)

- Πλεονεκτήματα
 - Κάνει μόνο $n - 1$ αντιμεταθέσεις

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΕΠΙΛΟΓΗ (6)

- **Μειονεκτήματα**
 - Ο χρόνος εκτέλεσης εξαρτάται ελάχιστα από την έκταση της υπάρχουσας ταξινόμησης σε ένα αρχείο
 - *Δηλαδή απαιτείται σχεδόν ίδιος χρόνος εκτέλεσης για την ταξινόμηση ενός αρχείου που είναι ήδη ταξινομημένο, ενός αρχείου που έχει όλα τα στοιχεία ίσα, και ενός αρχείου που έχει τυχαία ταξινόμηση*
 - Πολυπλοκότητα: $O(n^2)$

- <https://yongdanielliang.github.io/animation/web/SelectionSortNew.html>

ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (1)

- **Ταξινόμηση φυσαλίδας (Bubble sort)**
- Ένας από τους απλούστερους αλγόριθμους ταξινόμησης με την εξής λειτουργία
 - ξεκινάμε από την τελευταία θέση και συγκρίνουμε το τελευταίο στοιχείο με τα στοιχεία στα αριστερά του και αν είναι μικρότερο το αντιμεταθέτουμε, αλλιώς παίρνουμε το επόμενο στοιχείο στα αριστερά και κάνουμε την ίδια δουλειά μέχρι να τοποθετήσουμε το μικρότερο στοιχείο στην πρώτη θέση
 - ξεκινάμε από την τελευταία θέση και επαναλαμβάνουμε την ίδια διαδικασία για να τοποθετήσουμε το δεύτερο μικρότερο στοιχείο στη δεύτερη θέση

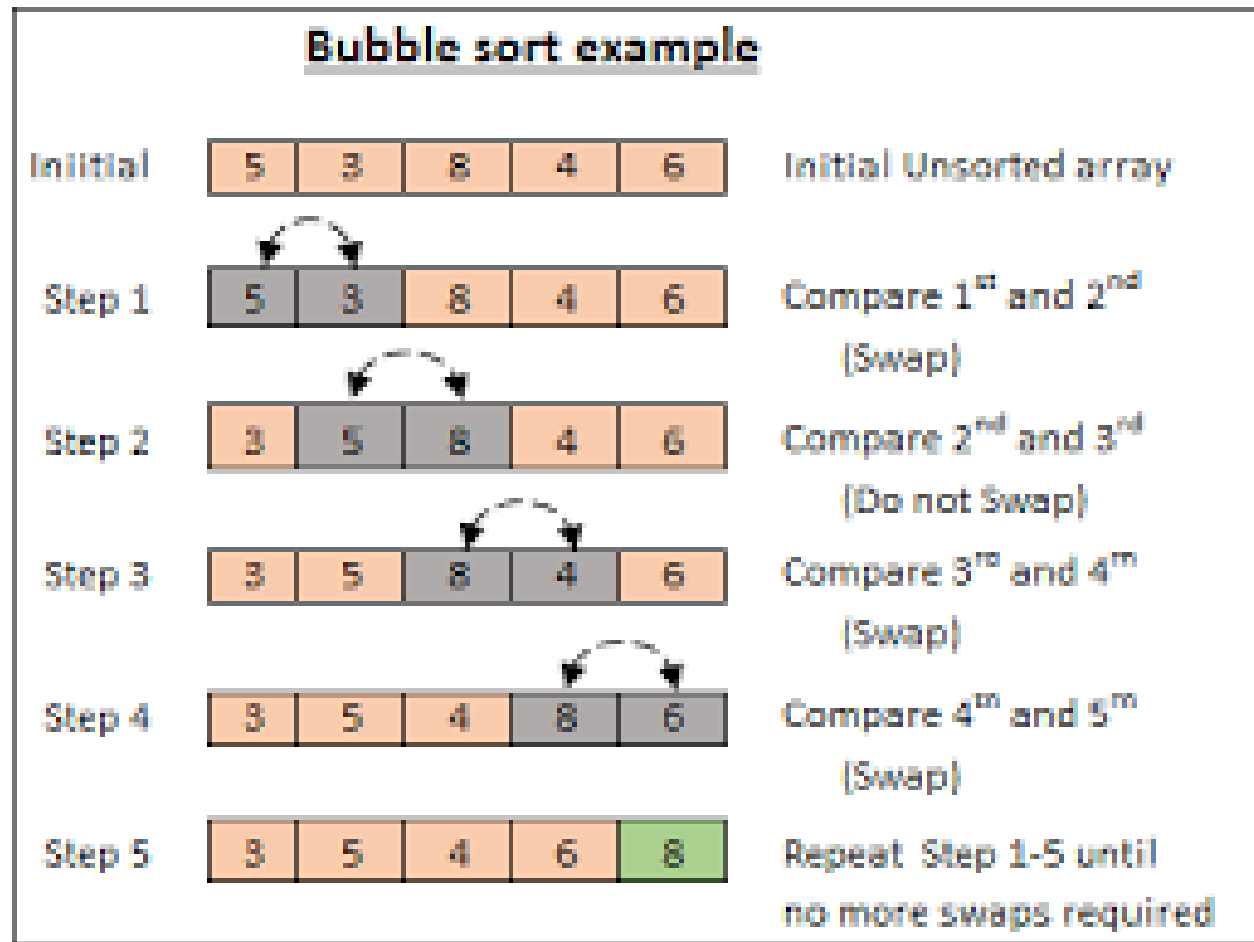
ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (1)

- Ταξινόμηση φυσαλίδας (Bubble sort)

Αλγόριθμος ταξινόμησης φυσαλίδας

1. Είσοδος: πίνακας A με n στοιχεία
2. Για i από 1 μέχρι $n-1$
 1. Για j από 1 μέχρι $n-i$
 1. Αν $A[j] > A[j+1]$ τότε
 1. Αντάλλαξε $A[j], A[j+1]$
 2. Τέλος_Αν
 2. Τέλος_Για
3. Τέλος_Για
4. Έξοδος: ταξινομημένος πίνακας A

ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (3)



ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (4)

- Πλεονεκτήματα
 - Εύκολη υλοποίηση

ΤΑΞΙΝΟΜΗΣΗ ΦΥΣΑΛΙΔΑΣ (5)

- **Μειονεκτήματα**
 - Αργότερη σε σύγκριση με την ταξινόμηση με εισαγωγή και την ταξινόμηση με επιλογή
 - Πολυπλοκότητα: $O(n^2)$

- <https://yongdanielliang.github.io/animation/web/BubbleSortNew.html>

ΣΥΓΚΡΙΣΗ ΣΤΟΙΧΕΙΩΔΩΝ ΑΛΓΟΡΙΘΜΩΝ ΤΑΞΙΝΟΜΗΣΗΣ (1)

- Οι χρόνοι εκτέλεσής τους διαφέρουν μόνο κατά έναν σταθερό παράγοντα, αλλά οι αλγόριθμοι λειτουργούν με αρκετά διαφορετικούς τρόπους
- Ο χρόνος εκτέλεσης των αλγόριθμων ταξινόμησης είναι ανάλογος με το πλήθος των **συγκρίσεων** που χρησιμοποιεί ο αλγόριθμος, το πλήθος των φορών που **αντιμετατίθενται** τα στοιχεία ή και με τις δύο αυτές παραμέτρους

ΣΥΓΚΡΙΣΗ ΣΤΟΙΧΕΙΩΔΩΝ ΑΛΓΟΡΙΘΜΩΝ ΤΑΞΙΝΟΜΗΣΗΣ (2)

- Κατά την ταξινόμηση με **επιλογή** γίνονται περίπου $\frac{n^2}{2}$ συγκρίσεις και n αντιμεταθέσεις
- Κατά την ταξινόμηση με **εισαγωγή** γίνονται κατά μέσο όρο περίπου $\frac{n^2}{4}$ συγκρίσεις και $\frac{n^2}{4}$ αντιμεταθέσεις, ενώ στη χειρότερη περίπτωση αυτά τα μεγέθη διπλασιάζονται
- Κατά την ταξινόμηση **φυσάλιδας** γίνονται περίπου $\frac{n^2}{2}$ συγκρίσεις και $\frac{n^2}{2}$ αντιμεταθέσεις, τόσο κατά μέσο όρο όσο και στη χειρότερη περίπτωση