

# Αλγόριθμοι Ταξινόμησης

## Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών  
Συστημάτων

Πανεπιστήμιο Αιγαίου

## Πρόβλημα Ταξινόμησης

- **Είσοδος** : ακολουθία  $n$  αριθμών ( $a_1, a_2, \dots, a_n$ ).
- **Έξοδος** : μετάθεση ( $a'_1, a'_2, \dots, a'_{n'}$ ) με στοιχεία σε **αύξουσα** σειρά ( $\forall i \ a'_i \leq a'_{i+1}$ ).
- Στατική συλλογή δεδομένων (όχι εισαγωγή και διαγραφή).
- **Αριθμοί** : κλειδιά εγγραφών με συσχεπζόμενη πληροφορία (*satellite data*).
  - Απαραίτητη ολική διάταξη.
  - Όμοια για κάθε σύνολο με ολική διάταξη.

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 2

## Ταξινόμηση

### □ Θεμελιώδες αλγορίθμικο πρόβλημα.

- Πολλές εφαρμογές (περ. 25% υπολογιστικού χρόνου).
- Ταχύτατη αναζήτηση σε ταξινομημένα δεδομένα.
- Σημαντικές αλγορίθμικές ιδέες και τεχνικές.

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 3

## Συγκριτικοί Αλγόριθμοι

- Αποκλειστικά με αποτελέσματα συγκρίσεων.
  - Δεν εκμεταλλεύονται τύπο δεδομένων (π.χ. αριθμούς, συμβολοσειρές) : γενική εφαρμογή.
  - **Ανάλυση** : αριθμός συγκρίσεων.  
αριθμός αντιμεταθέσεων.
  - Κάτω φράγμα #συγκρίσεων :  $\Omega(n \log n)$ .
  - Ταξινόμηση σε χρόνο  $\Theta(n+k)$  για φυσικούς αριθμούς στο  $\{1, \dots, k\}$  (π.χ. bin sort, radix sort).

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 4

## Μέθοδοι Ταξινόμησης

- **Αντιμετάθεση** κάθε ζεύγους στοιχείων εκτός διάταξης (bubble sort).
- **Εισαγωγή** στοιχείου σε κατάλληλη θέση ταξινομημένου πίνακα (insertion sort).
- **Επιλογή** μεγαλύτερου στοιχείου και τοποθέτηση στο τέλος (selection sort, heap sort).
- **Συγχώνευση** ταξινομημένων πινάκων : Διαίρεση στη μέση, ταξινόμηση, συγχώνευση (merge sort).
- **Διαιρεση** σε μικρότερα και μεγαλύτερα από στοιχείο-άξονα και ταξινόμηση (quicksort).

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 5

## Ταξινόμηση Φυσαλίδας

- **Αντιμετάθεση** κάθε ζεύγους εκτός διάταξης.
- Κάθε **επανάληψη** φέρνει μεγαλύτερο στοιχείο στο τέλος (δεξιά) («ανεβαίνει» στη φυσαλίδα).
- Διατηρεί δεξιά μεγαλύτερα στοιχεία ταξινομημένα.

```
for (i = n; i > 1; i--)  
    for (j = 0; j < i - 1; j++)  
        if (A[j] < A[j+1]) swap(A[j], A[j+1]);
```

$\begin{matrix} \xi & 7 & 2 & 1 & 8 & 3 & \epsilon & 4 \\ \xi & 2 & 1 & 7 & 3 & \epsilon & 4 & 8 \\ 2 & 1 & \xi & 3 & \epsilon & 4 & 7 & 8 \\ 1 & 2 & 3 & \xi & 4 & \epsilon & 7 & 8 \end{matrix}$	= 5
$\begin{matrix} 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \end{matrix}$	= 4
$\begin{matrix} 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \end{matrix}$	= 3
$\begin{matrix} 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \end{matrix}$	= 2

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 6

## Ταξινόμηση Φυσαλίδας

```
for (i = n; i > 1; i--)  
    for (j = 0; j < i - 1; j++)  
        if (A[j] < A[j+1]) swap(A[j], A[j+1]);
```

- **Ορθότητα** : Στο τέλος κάθε επανάληψης, επόμενο μεγαλύτερο στοιχείο στο τέλος (δεξιά).
- **Αριθμός συγκρίσεων** :  $\sum_{i=2}^n (i-1) = \sum_{i=1}^{n-1} i = n(n-1)/2 = \Theta(n^2)$
- **Χρόνος εκτέλεσης** :  $\Theta(n^2)$ .
- Όχι επιπλέον χώρος (in-place).
- **Στην πράξη**, σταματάμε αν ο πίνακας είναι ταξινομημένος.

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 7

## Ταξινόμηση με Εισαγωγή

- Διατηρεί ταξινομημένο αριστερό μέρος.
  - Εισάγει επόμενο στοιχείο σε κατάλληλη θέση.
- ```
for (i = 1; i < n; i++) {  
    key = A[i];  
    for (j = i-1; ((j >= 0) && (key < A[j])); j--)  
        A[j+1] = A[j];  
    A[j+1] = key;  
}
```

|                                                                                                                                                                                                    |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| $\begin{matrix} \xi & 7 & 2 & 1 & 8 & 3 & \epsilon & 4 \\ \xi & 7 & 2 & 1 & 8 & 3 & \epsilon & 4 \\ 2 & \xi & 7 & 1 & 8 & 3 & \epsilon & 4 \\ 1 & 2 & \xi & 7 & 8 & 3 & \epsilon & 4 \end{matrix}$ | = 4 |
| $\begin{matrix} 1 & 2 & \xi & 7 & 8 & 3 & \epsilon & 4 \\ 1 & 2 & \xi & 7 & 8 & 3 & \epsilon & 4 \\ 1 & 2 & \xi & 7 & 8 & 3 & \epsilon & 4 \\ 1 & 2 & \xi & 7 & 8 & 3 & \epsilon & 4 \end{matrix}$ | = 3 |
| $\begin{matrix} 1 & 2 & 3 & \xi & 7 & 8 & 3 & \epsilon \\ 1 & 2 & 3 & \xi & 7 & 8 & 3 & \epsilon \\ 1 & 2 & 3 & \xi & 7 & 8 & 3 & \epsilon \\ 1 & 2 & 3 & \xi & 7 & 8 & 3 & \epsilon \end{matrix}$ | = 2 |
| $\begin{matrix} 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \\ 1 & 2 & 3 & 4 & \xi & \epsilon & 7 & 8 \end{matrix}$ | = 1 |

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 8

## Ταξινόμηση με Εισαγωγή

- ```
for (i = 1; i < n; i++) {  
    key = A[i];  
    for (j = i-1; ((j >= 0) && (key < A[j])); j--)  
        A[j+1] = A[j];  
    A[i+1] = key; }  
  
□ Ορθότητα : τέλος επανάληψης  $i$ ,  $A[0 \dots j]$  ταξινομημένος.  
□ Χρόνος εκτέλεσης :  
    ■ Καλύτερη περ. : ταξινομημένος πίνακας,  $\Theta(n)$ .  
    ■ Χειρότερη περ. : πίνακας σε φθίνουσα σειρά,  $\sum_{i=1}^{n-1} (i-1) = \Theta(n^2)$   
    ■ Καλύτερη περ.  $\Omega(n)$ , χειρότερη περ.  $O(n^2)$ .  
□ 'Όχι επιπλέον χώρος (in-place).  
□ Μπορεί να γίνει με  $O(n \log n)$  συγκρίσεις;
```

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 9

## Ταξινόμηση με Επιλογή

- **Επιλογή** μεγαλύτερου στοιχείου και τοποθέτηση στο τέλος.
- Διατηρεί δεξιά μεγαλύτερα στοιχεία ταξινομημένα.

```
for (i = n; i > 1; i--) {  
    pos = 0;  
    for (j = 1; j < i; j++) // Find max(A[0...i-1])  
        if (A[pos] < A[j]) pos = j;  
    swap(A[pos], A[i-1]); }
```

ξ	7	2	1	8	3	6	4	= ξ
ξ	7	2	1	4	3	6	8	= ζ
ξ	6	2	1	4	3	7	8	= ζ
ξ	3	2	1	4	6	7	8	= 2

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 10

## Ταξινόμηση με Επιλογή

- ```
for (i = n; i > 1; i--) {  
    pos = 0;  
    for (j = 1; j < i; j++) // Find max(A[0...i-1])  
        if (A[pos] < A[j]) pos = j;  
    swap(A[pos], A[i-1]); }  
  
□ Ορθότητα : Στο τέλος κάθε επανάληψης, επόμενο μεγαλύτερο στοιχείο στο τέλος (δεξιά).  
□ Αριθμός συγκρίσεων :  $\sum_{i=2}^n (i-1) = \sum_{i=1}^{n-1} i = n(n-1)/2 = \Theta(n^2)$   
□ Χρόνος εκτέλεσης :  $\Theta(n^2)$ .  
□ 'Όχι επιπλέον χώρος (in-place).  
□ Στην πράξη, σταματάμε αν ο πίνακας είναι ταξινομημένος.
```

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 11

## Ταξινόμηση με Ουρά Προτεραιότητας

- **Αρχικοποίηση** : δημιουργησε ουρά προτεραιότητας με όλα τα στοιχεία.
- **Επιλογή** (εισαγωγή) **μέγιστου** και τοποθέτηση στο τέλος (μεχρι ταξινόμηση).
- **Απόδοση** εξαρτάται από ουρά προτεραιότητας.
- Ταξινόμηση με **Εισαγωγή** :
  - ΟΠ : γραμμική λίστα (πίνακας) σε αύξουσα σειρά.
  - Χρόνος εισαγωγής :  $O(1) + O(2) + \dots + O(n) = O(n^2)$ .
  - Δεν χρειάζεται επιλογή!
- Ταξινόμηση με **Επιλογή** :
  - ΟΠ : γραμμική λίστα (πίνακας) όχι ταξινομημένη.
  - Δεν χρειάζεται εισαγωγή!
  - Χρόνος επιλογής :  $O(n) + O(n-1) + \dots + O(1) = O(n^2)$ .

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 12

## Heap-Sort

- **Αρχικοποίηση** : δημιουργησε σωρό με όλα τα στοιχεία.
  - `createHeap()` : χρόνος  $\Theta(n)$ .
- **Εξαγωγή μέγιστου** και τοποθέτηση στο τέλος ( $n - 1$  φορές).
  - `deleteMax()` : χρόνος  $\Theta(\log n)$ .
- **Χρόνος** :  $\Theta(n) + n \Theta(\log n) = \Theta(n \log n)$ .

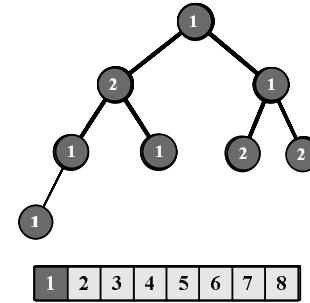
```
hs = n;
createHeap(A, n);
for (i = n; i > 1; i--)
{
    swap(A[1], A[i]); hs--;
    fixHeap(A, 1);
}
```

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 13

## Heap-Sort : Παράδειγμα

```
createHeap(A, n);
for (i = n; i > 1; i--) {
    swap(A[1], A[i]); hs--;
    fixHeap(A, 1); }
```



Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 14

## Σύνοψη

| Αλγόριθμος | Καλύτερη      | Μέση          | Χειρότερη     | Χώρος  |
|------------|---------------|---------------|---------------|--------|
| BubbleS    | $\Omega(n)$   | $O(n^2)$      | $O(n^2)$      | $O(1)$ |
| InsertionS | $\Omega(n)$   | $O(n^2)$      | $O(n^2)$      | $O(1)$ |
| SelectionS | $\Omega(n)$   | $O(n^2)$      | $O(n^2)$      | $O(1)$ |
| HeapS      | $O(n \log n)$ | $O(n \log n)$ | $O(n \log n)$ | $O(1)$ |

Δομές Δεδομένων

Αλγόριθμοι Ταξινόμησης 15