

# Ταξινόμηση με Διαίρει-και-Βασίλευε

Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών  
Συστημάτων

Πανεπιστήμιο Αιγαίου

## Διαίρει-και-Βασίλευε

- Γενική μέθοδος σχεδιασμού αλγορίθμων:
  - **Διαίρεση** σε ( $\geq 2$ ) υπο-προβλήματα (σημαντικά) μικρότερου μεγέθους.
  - **Ανεξάρτητη** (αναδρομική) επίλυση υπο-προβλημάτων (για μικρά υπο-προβλήματα εφαρμόζουμε στοιχειώδεις αλγορίθμους).
  - **Σύνθεση** λύσης αρχικού προβλήματος από λύσεις υπο-προβλημάτων.
- Ισχυρή μέθοδος, πολλές σημαντικές εφαρμογές!
- (Εύκολη) ανάλυση με **αναδρομικές εξισώσεις**.
- **Ταξινόμηση** : merge-sort, quicksort.

Δομές Δεδομένων

Ταξινόμηση με Διάρει-και-Βασίλευε 2

## Ταξινόμηση με Συγχώνευση

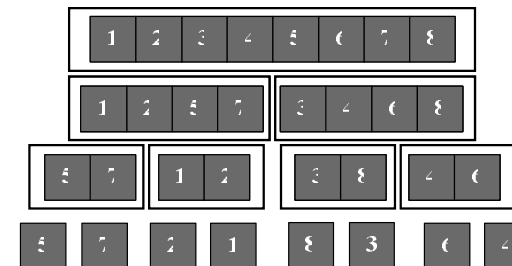
- **Διαίρεση** ακολουθίας εισόδου ( $n$  στοιχεία) σε δύο υπο-ακολουθίες ίδιου μήκους ( $n/2$  στοιχεία).
- **Ταξινόμηση** υπο-ακολουθιών **αναδρομικά**.
- **Συγχώνευση** ταξινομημένων υπο-ακολουθιών σε μία ταξινομημένη ακολουθία.

```
mergeSort(int A[], int left, int right) {  
    if (left >= right) return; // At most 1 element  
    mid = (left + right) / 2;  
    mergeSort(A, left, mid);  
    mergeSort(A, mid+1, right);  
    // merge A[left...mid] and A[mid+1...right]  
    // into A[left...right]  
    merge(A, left, mid, right); }
```

Δομές Δεδομένων

Ταξινόμηση με Διάρει-και-Βασίλευε 3

## Ταξινόμηση με Συγχώνευση

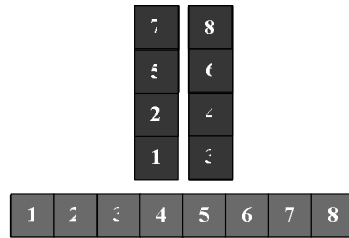


Δομές Δεδομένων

Ταξινόμηση με Διάρει-και-Βασίλευε 4

## Συγχώνευση

- **Συγχώνευση** ταξινομημένων  $A[low \dots mid]$  και  $A[mid+1 \dots up]$  σε ταξινομημένο  $A[low \dots up]$ .



Δομές Δεδομένων

Ταξινόμηση με Διάιρα-και-Βασίλευε 5

## Συγχώνευση

- **Συγχώνευση** ταξινομημένων  $A[low \dots mid]$  και  $A[mid+1 \dots up]$  σε ταξινομημένο  $A[low \dots up]$ .
  - $X[up - low + 1] \leftarrow A[low \dots up]$ ; // προσωρινή αποθήκευση
  - $i : 0 \leq i \leq xmid$  // δείκτης αριστερό τμήμα
  - $j : xmid+1 \leq j \leq xup$  // δείκτης δεξιό τμήμα
  - $k : low \leq k \leq up$  // δείκτης στο συγχωνευμένο πίνακα
- $X[i]$  : μικρότερο διαθέσιμο στοιχείο στο αριστερό τμήμα.
- $X[j]$  : μικρότερο διαθέσιμο στοιχείο στο δεξιό τμήμα.
- Μικρότερο από  $X[i]$ ,  $X[j]$  αντιγράφεται σε θέση  $A[k]$ .
- Όταν ένα τμήμα **εξαντληθεί**, αντιγράφουμε όλα τα στοιχεία του άλλου στο  $A[ ]$ .

Δομές Δεδομένων

Ταξινόμηση με Διάιρα-και-Βασίλευε 6

## Ορθότητα

- Ορθότητα **merge** επειδή τα τμήματα είναι ταξινομημένα.
  - Όταν ένα στοιχείο μεταφέρεται στον  $A[ ]$ , δεν υπάρχει μικρότερο διαθέσιμο στοιχείο στα δύο τμήματα.
- Ορθότητα **mergeSort** αποδεικνύεται επαγωγικά :
  - Βάση (ένα στοιχείο) τετριμμένη.
  - Δύο τμήματα σωστά ταξινομημένα (επαγωγική υποθ.) και συγχωνεύονται σωστά (ορθότητα merge)  $\Rightarrow$  Σωστά ταξινομημένος πίνακας  $A[ ]$ .

Δομές Δεδομένων

Ταξινόμηση με Διάιρα-και-Βασίλευε 7

## Χρόνος Εκτέλεσης

- Χρόνος εκτέλεσης **merge** ( $n$  στοιχεία) :  $\Theta(n)$  (γραμμικός)
  - $\Theta(1)$  λειτουργίες για κάθε μεταφορά στοιχείου στον  $A[ ]$ .
- Χρόνος εκτελ. αναδρομικών αλγ. με διατύπωση και λύση αναδρομικής εξίσωσης λειτουργίας.
- **$T(n)$**  : χρόνος για ταξινόμηση  $n$  στοιχείων.
  - **$T(n/2)$**  : ταξινόμηση αριστερού τμήματος ( $n/2$  στοιχεία).
  - **$T(n/2)$**  : ταξινόμηση δεξιού τμήματος ( $n/2$  στοιχεία).
  - **$\Theta(n)$**  : συγχώνευση ταξινομημένων τμημάτων.
$$T(n) = 2T(n/2) + \Theta(n), T(1) = \Theta(1)$$
- Χρόνος εκτέλεσης **mergeSort** :  **$T(n) = \Theta(n \log n)$** .

Δομές Δεδομένων

Ταξινόμηση με Διάιρα-και-Βασίλευε 8

## Δέντρο Αναδρομής

$$T(n) = 2T(n/2) + \Theta(n),$$

$$T(1) = \Theta(1)$$

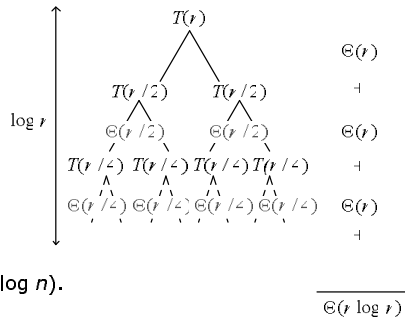
### Δέντρο αναδρομής :

Ύψος :  $\Theta(\log n)$

# κορυφών :  $\Theta(n)$

Χρόνος / επίπεδο :  $\Theta(n)$

**Συνολικός χρόνος :**  $\Theta(n \log n)$ .



Δομές Δεδομένων

Ταξινόμηση με Διάρρα-και-Βασίλευε 9

## Σύνοψη

### □ Ταξινόμηση με συγχώνευση (merge-sort):

- Χρόνος εκτέλεσης :  $\Theta(n \log n)$  – βέλτιστος.
- Χώρος :  $\Theta(n)$  – όχι in-place.
- Όχι πολύ γρήγορη στην πράξη.
- Αναδρομή καθυστερεί και απαιτεί μνήμη.
  - Επαναληπτική υλοποίηση.
- Ακολουθιακή προσπέλαση στη συγχώνευση!
  - Κατάλληλη για πολύ μεγάλο αριθμό στοιχείων σε δευτερεύουσα μνήμη (π.χ. σκληρό δίσκο).

Δομές Δεδομένων

Ταξινόμηση με Διάρρα-και-Βασίλευε 10

## Σύνοψη

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

Δομές Δεδομένων

Ταξινόμηση με Διάρρα-και-Βασίλευε 11