

# ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

## ΜΑΘΗΜΑ 9

Ηλίας Κουτσουπιάς

Τμήμα Πληροφορικής και Τηλεπικοινωνιών  
Πανεπιστήμιο Αθηνών  
elias@@di.uoa.gr

06/04/2009

ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΤΑΞΙΝΟΜΗΣΗΣ Δίνεται ακολουθία  
ακεραίων  $a_0, \dots, a_{n-1}$ . Να ταξινομηθεί σε  
αύξουσα σειρά.

## ΠΑΡΑΔΕΙΓΜΑ

- Είσοδος:  $a = [7, 1, 2, 6, 9, 4, 6]$
- Έξοδος:  $a = [1, 2, 4, 6, 6, 7, 9]$

## ΑΛΓΟΡΙΘΜΟΙ

**INSERTION SORT** Για να τα ταξινομήσουμε την ακολουθία με  $n$  στοιχεία, ταξινομούμε τα πρώτα  $n - 1$  στοιχεία αναδρομικά, και μετά εισάγουμε το τελευταίο στοιχείο στην κατάλληλη θέση.

**MERGESORT** Χώρισε την ακολουθία σε δυο ίσα μέρη. Ταξινόμησε κάθε μέρος. Συγχώνευσε.

**QUICKSORT** Με βάση την τιμή ενός στοιχείου διαμέρισε (partition) την ακολουθία σε στοιχεία μικρότερης τιμής και σε στοιχεία μεγαλύτερης τιμής. Ταξινόμησε ξεχωριστά τα στοιχεία της μικρότερης τιμής και της μεγαλύτερης τιμής.

QUICKSORT

- Για να αναλύσουμε τον αναδρομικό αλγόριθμο, έστω  $T(n)$  ο χρόνος για να ταξινομηθούν  $n$  στοιχεία.



$$T(n) = T(n - 1) + n,$$

επειδή

- $T(n - 1)$  είναι ο χρόνος για να ταξινομηθούν τα πρώτα  $n - 1$  στοιχεία
- $O(n)$  είναι ο χρόνος για να εισαχθεί το τελευταίο στοιχείο

Η λύση της αναδρομικής εξίσωσης είναι

$$T(n) = \Theta(n^2)$$

- Για να αναλύσουμε τον αναδρομικό αλγόριθμο, έστω  $T(n)$  ο χρόνος για να ταξινομηθούν  $n$  στοιχεία.



$$T(n) = 2T\left(\frac{n}{2}\right) + n,$$

επειδή

- $T(n/2)$  είναι ο χρόνος για να ταξινομηθούν τα μισά στοιχεία
- $O(n)$  είναι ο χρόνος για να συγχωνευθούν τα  $n$  στοιχεία

Η λύση της αναδρομικής εξίσωσης είναι

$$T(n) = \Theta(n \log n)$$

# ΑΝΑΛΥΣΗ ΤΗΣ QUICKSORT - ΧΕΙΡΟΤΕΡΗ ΠΕΡΙΠΤΩΣΗ

- Για να αναλύσουμε τον αναδρομικό αλγόριθμο, έστω  $T(n)$  ο χρόνος για να ταξινομηθούν  $n$  στοιχεία.



$$T(n) = T(n-1) + T(1) + n,$$

επειδή

- Υποθέτουμε πως η ακολουθία χωρίζεται σε δυο μέρη με  $n-1$  και  $1$  στοιχείο.
- $T(n-1)$  και  $T(1)$  είναι ο χρόνος για να ταξινομηθούν τα δυο μέρη
- $O(n)$  είναι ο χρόνος για να διαμεριστούν τα  $n$  στοιχεία

Η λύση της αναδρομικής εξίσωσης είναι

$$T(n) = \Theta(n^2)$$

```
def quicksort (a, i, j):  
    """Sorts the elements of array a[i:j] using partition"""  
    if j-i>1:  
        m = partition(a[i], a, i+1, j)  
        a[i], a[m-1] = a[m-1], a[i]  
        quicksort(a, i, m-1)  
        quicksort(a, m, j)
```

# ΑΝΑΛΥΣΗ ΤΗΣ QUICKSORT - ΚΑΛΥΤΕΡΗ ΠΕΡΙΠΤΩΣΗ

- Για να αναλύσουμε τον αναδρομικό αλγόριθμο, έστω  $T(n)$  ο χρόνος για να ταξινομηθούν  $n$  στοιχεία.



$$T(n) = 2T\left(\frac{n}{2}\right) + n,$$

επειδή

- Υποθέτουμε πως η ακολουθία χωρίζεται σε δυο ίσα μέρη.
- $2T(n/2)$  είναι ο χρόνος για να ταξινομηθούν τα δυο μέρη
- $O(n)$  είναι ο χρόνος για να διαμεριστούν τα  $n$  στοιχεία

Η λύση της αναδρομικής εξίσωσης είναι

$$T(n) = \Theta(n \log n)$$



# ΑΝΑΛΥΣΗ ΤΗΣ ΠΙΘΑΝΟΤΙΚΗΣ QUICKSORT

- Για να αναλύσουμε τον αναδρομικό αλγόριθμο, έστω  $T(n)$  ο χρόνος για να ταξινομηθούν  $n$  στοιχεία.

$$\begin{aligned}T(n) &= \frac{1}{n}(T(0) + T(n-1)) \\&\quad + \frac{1}{n}(T(1) + T(n-2)) + \\&\quad + \dots \\&\quad + \frac{1}{n}(T(n-1) + T(0)) + n \\&= \frac{2}{n}(T(0) + T(1) + \dots + T(n-1)) + n,\end{aligned}$$

επειδή

- Υποθέτουμε πως η ακολουθία χωρίζεται σε δυο τυχαία μέρη.
- $O(n)$  είναι ο χρόνος για να διαμεριστούν τα  $n$  στοιχεία

Η λύση της αναδρομικής εξίσωσης είναι

$$T(n) = \Theta(n \log n)$$