



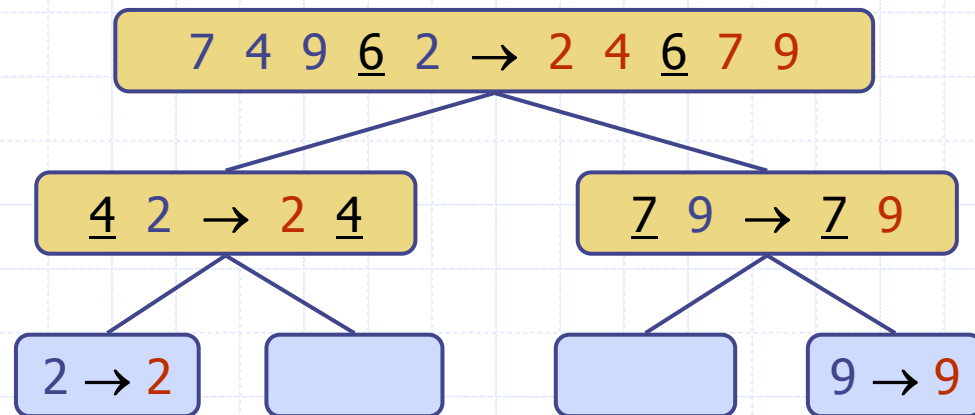
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Αλγόριθμοι και πολυπλοκότητα

## Ταχυταξινόμηση (Quick-Sort)

Ιωάννης Τόλλης  
Τμήμα Επιστήμης Υπολογιστών

# Ταχυταξινόμηση (Quick-Sort)



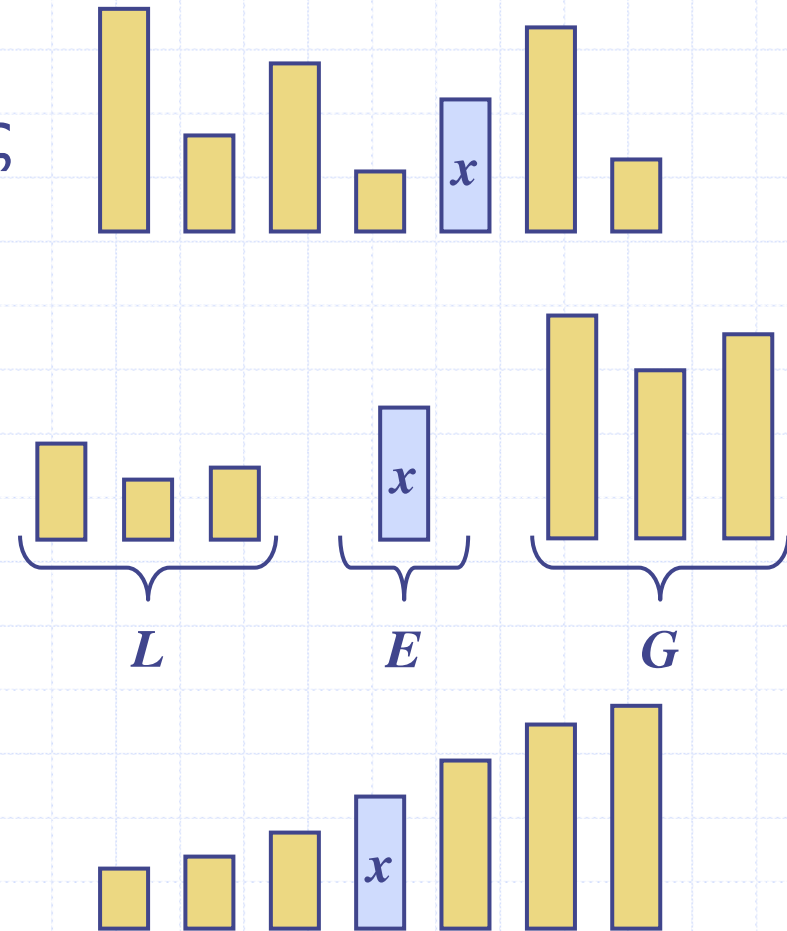
# Κύρια σημεία και Διάβασμα

- ◆ Ταχυταξινόμηση (Quick-sort)
  - Αλγόριθμος
  - Βήμα διαμέρισης
  - Δένδρο ταχυταξινόμησης
  - Παράδειγμα εκτέλεσης
- ◆ Ανάλυση της ταχυταξινόμησης
- ◆ Ταχυταξινόμηση με αναδιάταξη
- ◆ Σύνοψη αλγορίθμων ταξινόμησης

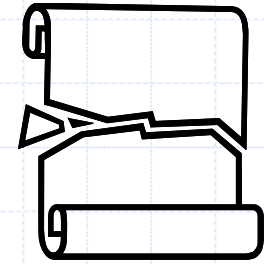
# Ταχυταξινόμηση

❖ **Ταχυταξινόμηση:** αλγόριθμος ταξινόμησης, δημιουργημένος από συνθήκες τυχειότητας, βασισμένος στην τεχνική «διαίρει και κυρίευε»:

- **Διαίρει:** επέλεξε τυχαίο στοιχείο  $x$  (οδηγός) και διαίρεσε το  $S$  σε
  - ♦  $L$  στοιχεία μικρότερα του  $x$
  - ♦  $E$  στοιχεία ίσα με το  $x$
  - ♦  $G$  στοιχεία μεγαλύτερα του  $x$
- **Επανάληψη:** ταξινόμηση τα  $L$  και  $G$
- **Κυρίευε:** ένωσε τα  $L$ ,  $E$  και  $G$



# Διαμέριση



- ◆ Διαιρούμε μια ακολουθία εισόδου ως εξής:
  - Αφαιρούμε, με τη σειρά, κάθε στοιχείο  $y$  από το  $S$  και
  - Εισάγουμε το  $y$  στο  $L$ ,  $E$  ή  $G$ , ανάλογα με το αποτέλεσμα της σύγκρισης με το  $x$  (οδηγός)
- ◆ Κάθε εισαγωγή και αφαίρεση γίνεται στην αρχή ή το τέλος της ακολουθίας, και έτσι χρειάζεται  $O(1)$  χρόνο
- ◆ Έτσι, το βήμα διαμέρισης της ταχυσταξινόμησης χρειάζεται  $O(n)$  χρόνο

**Algorithm** *partition*( $S, p$ )

**Input** sequence  $S$ , position  $p$  of pivot

**Output** subsequences  $L, E, G$  of the elements of  $S$  less than, equal to, or greater than the pivot, resp.

$L, E, G \leftarrow$  empty sequences

$x \leftarrow S.remove(p)$

**while**  $\neg S.isEmpty()$

$y \leftarrow S.remove(S.first())$

**if**  $y < x$

$L.insertLast(y)$

**else if**  $y = x$

$E.insertLast(y)$

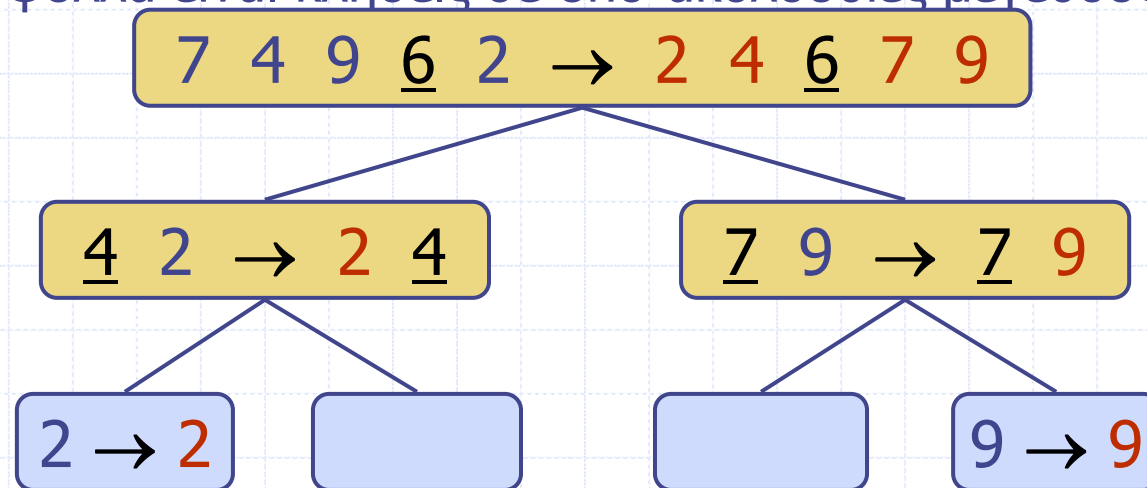
**else** {  $y > x$  }

$G.insertLast(y)$

**return**  $L, E, G$

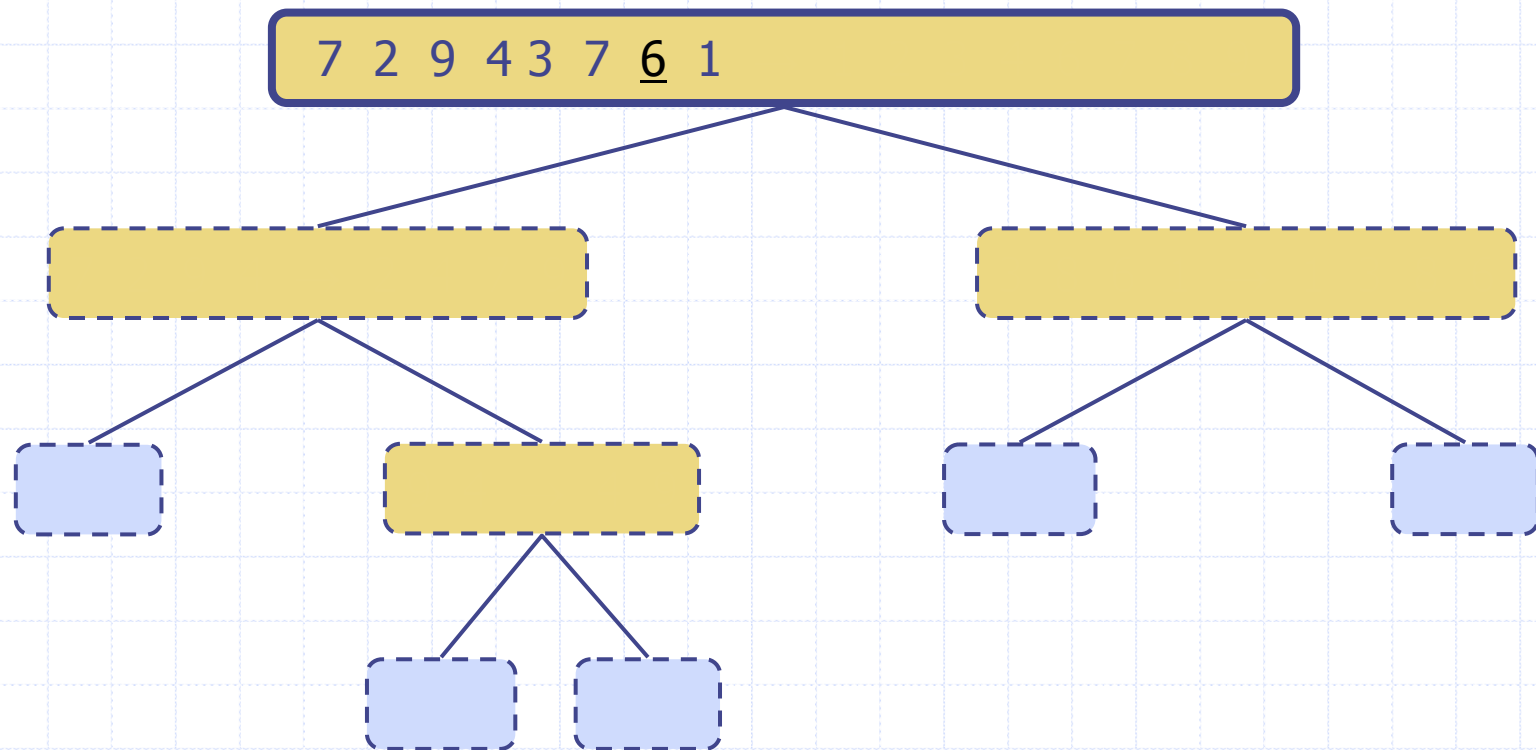
# Δένδρο ταχυταξινόμησης

- ◆ Κάθε εκτέλεση της ταχυταξινόμησης απεικονίζεται με ένα δυαδικό δένδρο
  - Κάθε κόμβος αναπαριστά μία αναδρομική κλήση της ταχυταξινόμησης και αποθηκεύει
    - ◆ Μη ταξινομημένη ακολουθία πριν την εκτέλεση και τον οδηγό της
    - ◆ Ταξινομημένη ακολουθία στο τέλος της εκτέλεσης
  - Η ρίζα είναι η αρχική κλήση
  - Τα φύλλα είναι κλήσεις σε υπό-ακολουθίες μεγέθους 0 ή 1



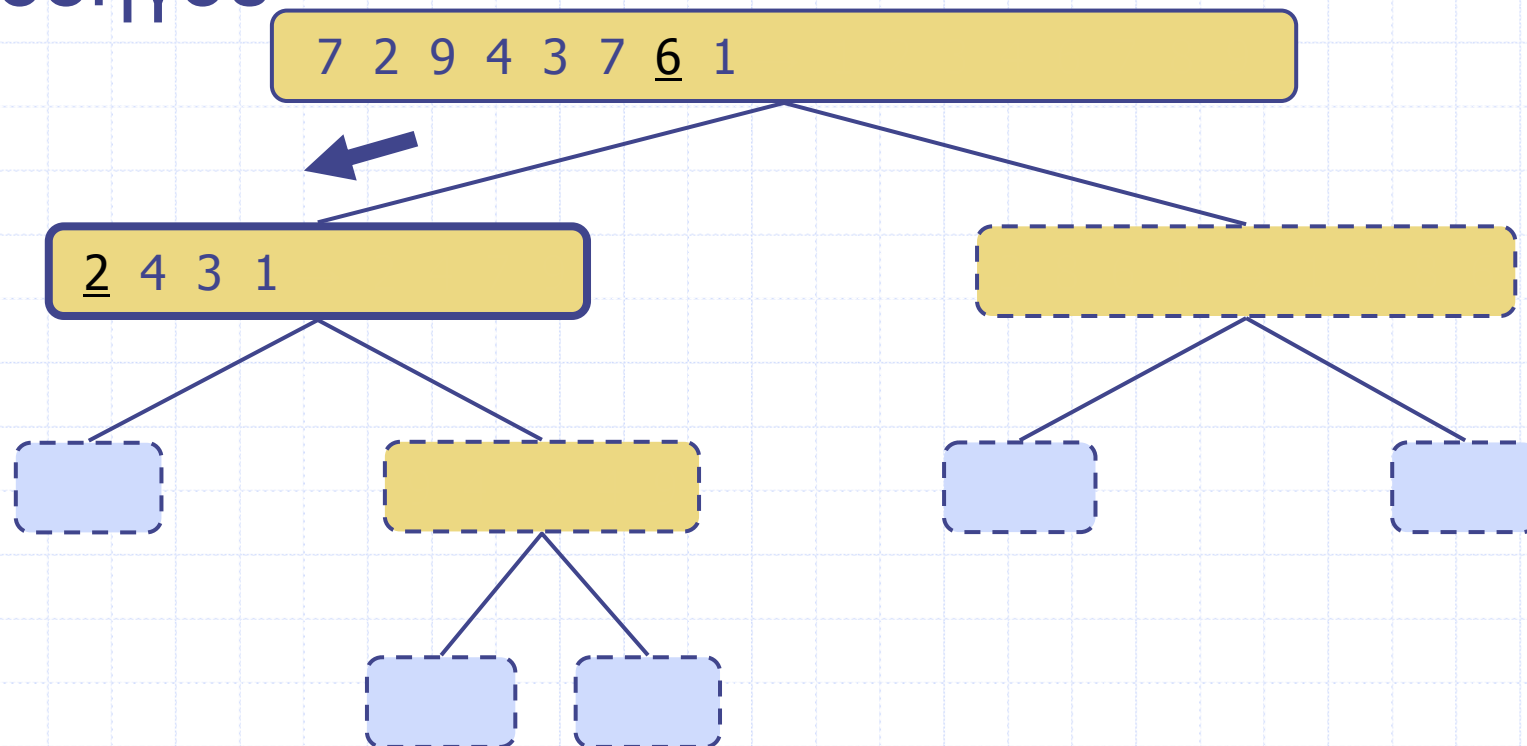
# Παράδειγμα Εκτέλεσης

## ◆ Επιλογή του οδηγού



# Παράδειγμα Εκτέλεσης (συν.)

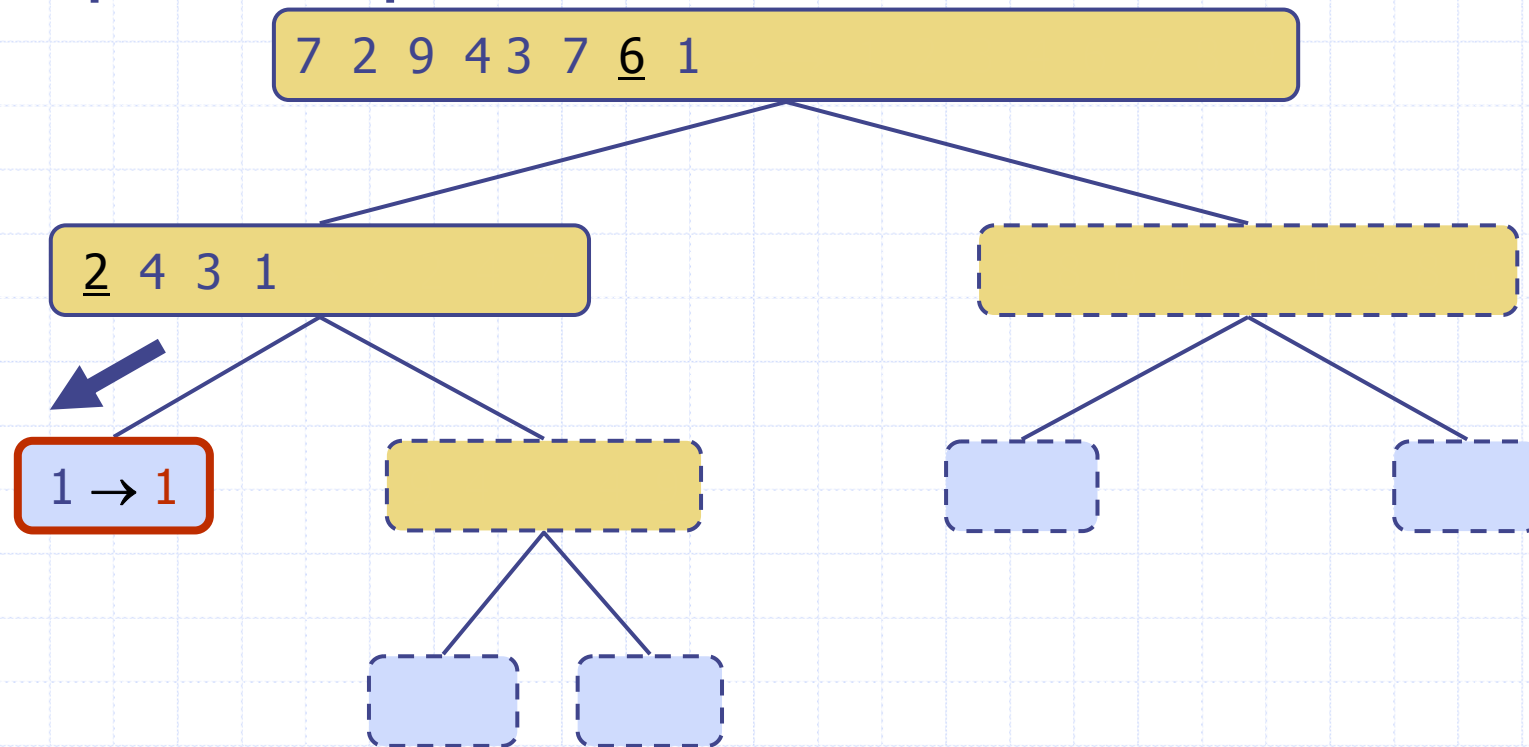
- ◆ Διαμέριση, αναδρομική κλήση, επιλογή οδηγού





# Παράδειγμα Εκτέλεσης (συν.)

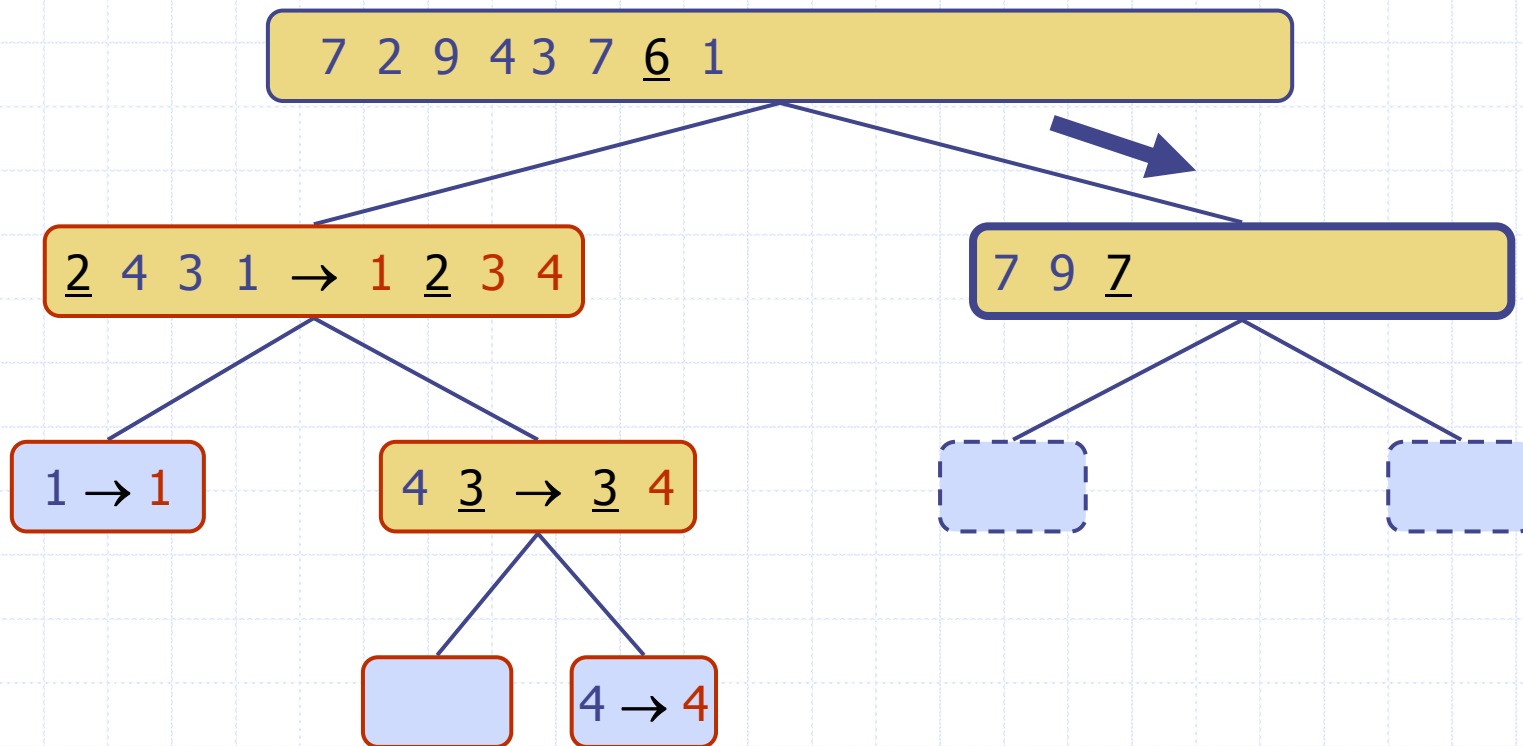
- ◆ Διαμέριση, αναδρομική κλήση, βασική περίπτωση





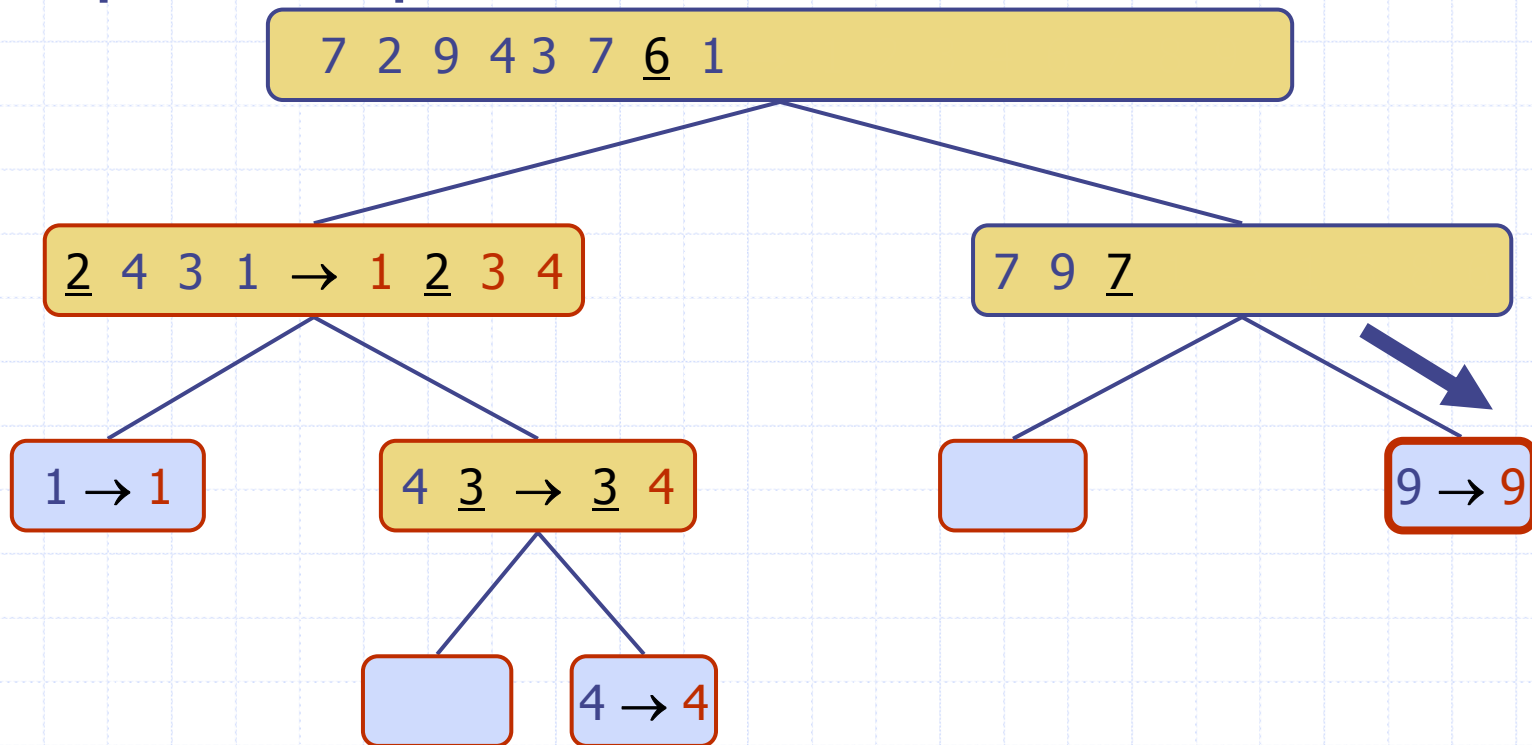
# Παράδειγμα Εκτέλεσης (συν.)

◆ Αναδρομική κλήση, επιλογή οδηγού



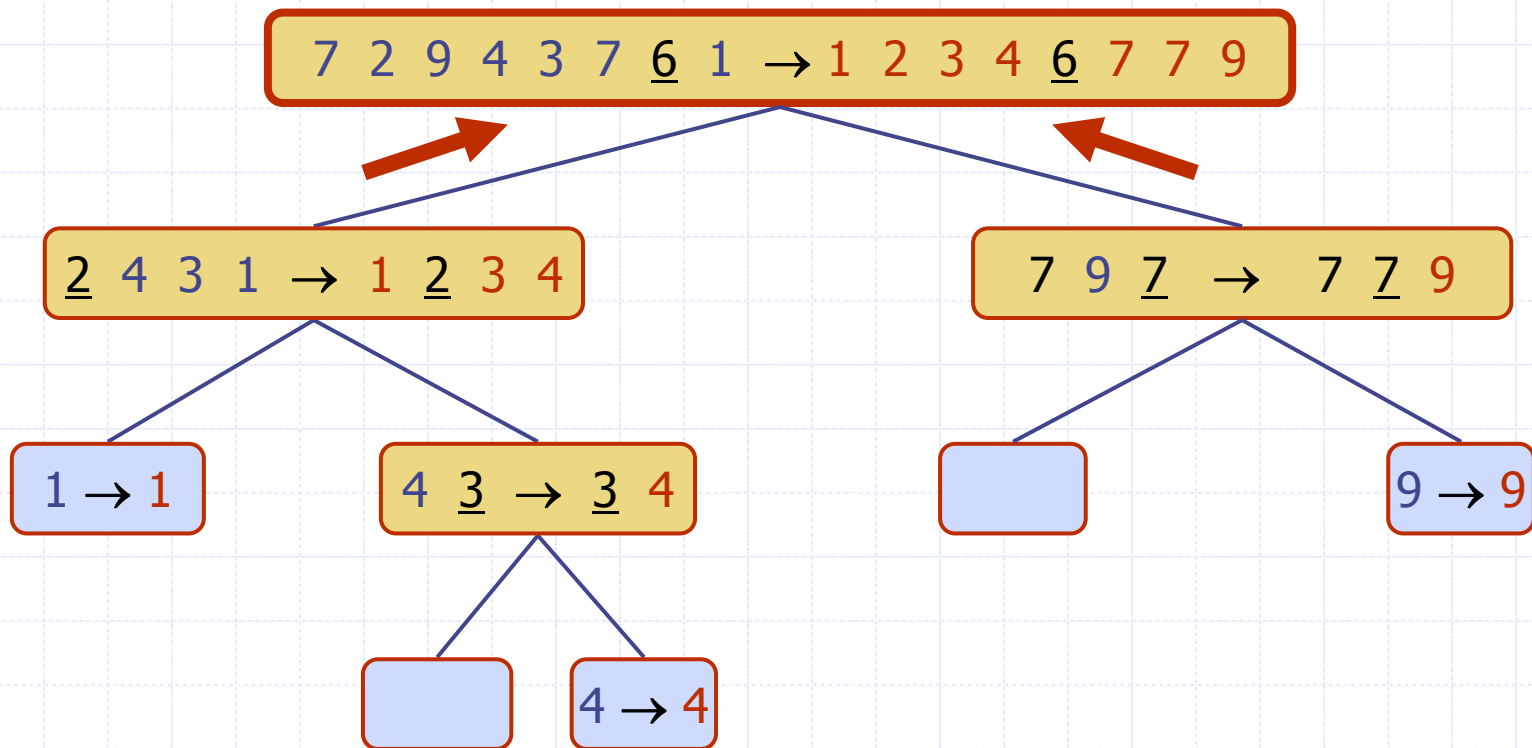
# Παράδειγμα Εκτέλεσης (συν.)

- ◆ Διαμέριση, ..., αναδρομική κλήση, βασική περίπτωση



# Παράδειγμα Εκτέλεσης (συν.)

◆ Ένωση, ένωση



# Χρόνος Εκτέλεσης Χειρότερης Περίπτωσης

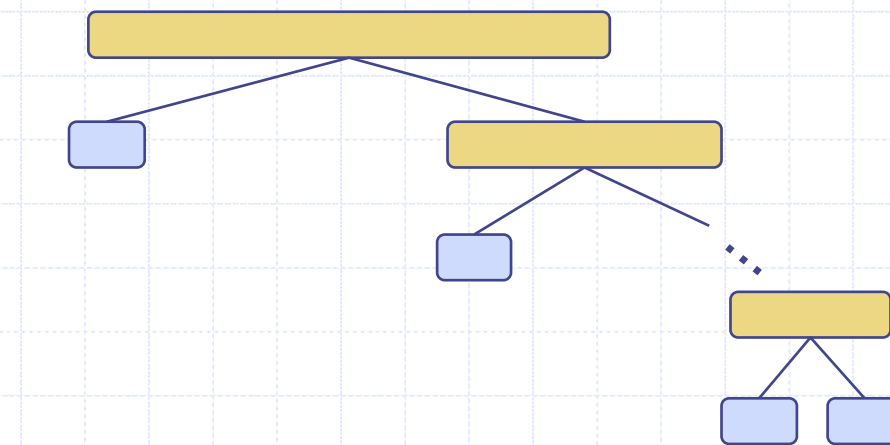
- ◆ Η χειρότερη περίπτωση της ταχυταξινόμησης συμβαίνει όταν το στοιχείο οδηγός είναι το μοναδικό ελάχιστο ή μέγιστο στοιχείο
- ◆ Ένα από τα  $L$  και  $G$  έχει μέγεθος  $n - 1$  στοιχείο και το άλλο 0
- ◆ Ο χρόνος εκτέλεσης είναι ανάλογος του αθροίσματος

$$n + (n - 1) + \dots + 2 + 1$$

- ◆ Έτσι, ο χρόνος εκτέλεσης της χειρότερης περίπτωσης της ταχυταξινόμησης :  $O(n^2)$

βάθος χρόνος

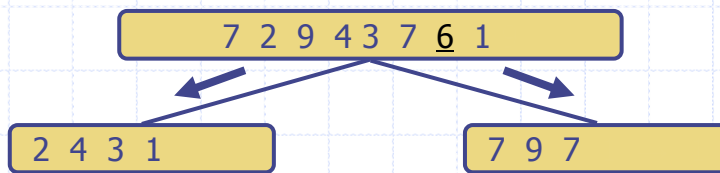
0	$n$
1	$n - 1$
...	...
$n - 1$	1



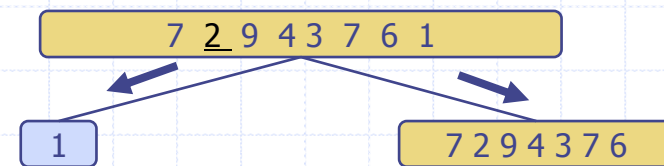
Ταχυταξινόμηση

# Αναμενόμενος Χρόνος Εκτέλεσης

- ◆ Έστω αναδρομική κλήση της ταχυσταξινόμησης σε μια ακολουθία μεγέθους  $s$ 
  - **Καλή κλήση:** τα μεγέθη των  $L$  και  $G$  είναι το καθένα μικρότερα του  $3s/4$
  - **Κακή κλήση:** ένα από τα  $L$  και  $G$  έχουν μέγεθος μεγαλύτερο του  $3s/4$

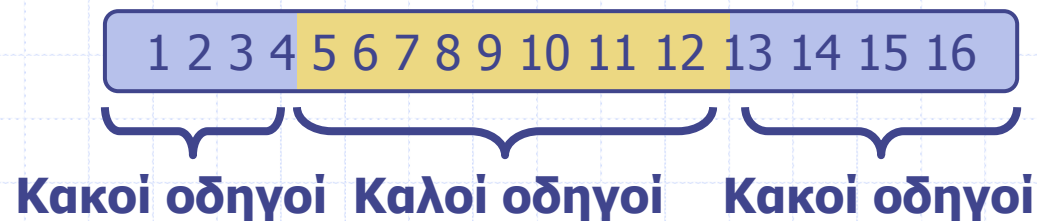


Καλή κλήση



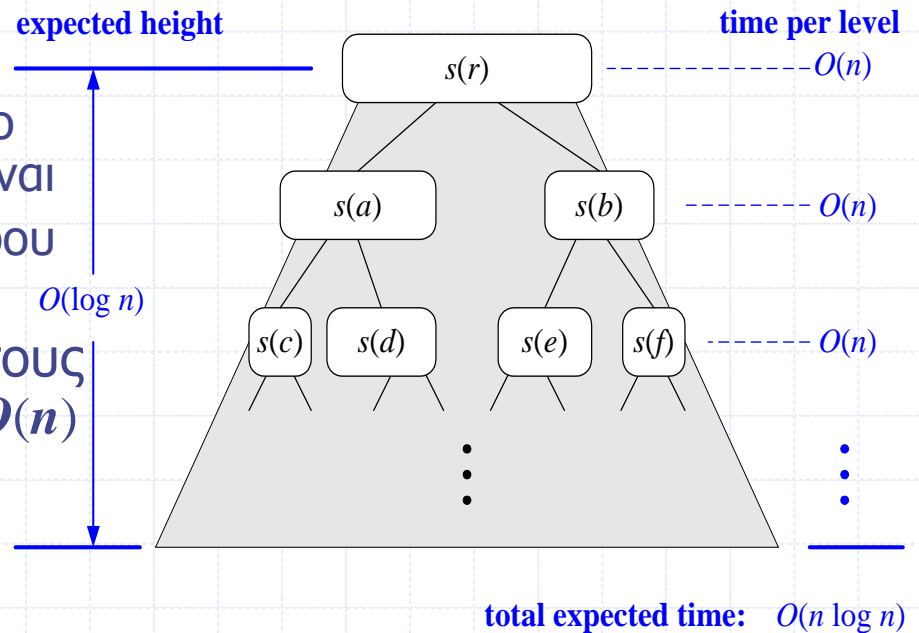
Κακή κλήση

- ◆ Μια κλήση είναι **καλή** με πιθανότητα  $1/2$ 
  - $1/2$  των πιθανών οδηγών προκαλούν καλές κλήσεις:



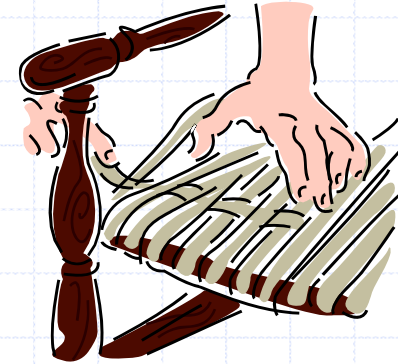
# Αναμενόμενος Χρόνος Εκτέλεσης, Μέρος 2

- ◆ **Πιθανοτικό Γεγονός:** Ο αναμενόμενος αριθμός ρίψεων κέρματος για να έρθουν  $k$  κορώνες είναι  $2k$
- ◆ Για ένα κόμβο βάθους  $i$ , περιμένουμε
  - $i/2$  προγόνους να είναι καλές κλήσεις
  - Το μέγεθος της ακολουθίας εισόδου για την τρέχουσα κλήση είναι το πολύ  $(3/4)^{i/2}n$
- ◆ Έτσι, έχουμε
  - Για ένα κόμβο βάθους  $2\log_{4/3}n$ , το αναμενόμενο μέγεθος εισόδου είναι
  - Το αναμενόμενο ύψος του δένδρου quick-sort tree είναι  $O(\log n)$
- ◆ Το ποσό δουλειάς που έγινε για τους κόμβους του ίδιου βάθους είναι  $O(n)$
- ◆ Έτσι, αναμενόμενος χρόνος εκτέλεσης ταχυταξινόμησης:  $O(n \log n)$





# Ταχυταξινόμηση με αναδιάταξη



- ◆ Η ταχυταξινόμηση μπορεί να υλοποιηθεί με αναδιάταξη στοιχείων
- ◆ Στο βήμα κατάρτησης, χρησιμοποιούμε λειτουργία αντικατάστασης για να επανατοποθετήσουμε τα στοιχεία της ακολουθίας εισόδου, ώστε:
  - τα στοιχεία μικρότερα του οδηγού έχουν βαθμό κατάταξης μικρότερο του  $h$
  - τα στοιχεία ίσα με τον οδηγό έχουν βαθμό ανάμεσα στα  $h$  και  $k$
  - τα στοιχεία μεγαλύτερα του οδηγού έχουν βαθμό μεγαλύτερο του  $k$
- ◆ Οι αναδρομικές κλήσεις θεωρούν:
  - Στοιχεία με βαθμό  $< h$
  - Στοιχεία με βαθμό  $> k$

**Algorithm** *inPlaceQuickSort*( $S, l, r$ )

**Input** sequence  $S$ , ranks  $l$  and  $r$

**Output** sequence  $S$  with the elements of rank between  $l$  and  $r$  rearranged in increasing order

**if**  $l \geq r$

**return**

$i \leftarrow$  a random integer between  $l$  and  $r$

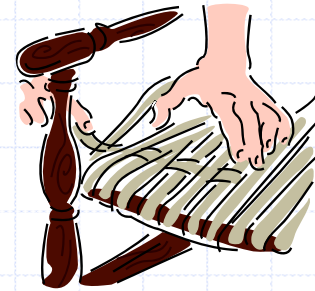
$x \leftarrow S.elemAtRank(i)$

$(h, k) \leftarrow inPlacePartition(x)$

*inPlaceQuickSort*( $S, l, h - 1$ )

*inPlaceQuickSort*( $S, k + 1, r$ )

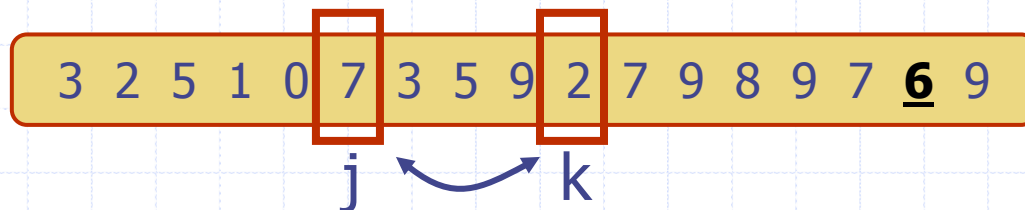
# Διαμέριση με αναδιάταξη



- ◆ Εκτέλεση την κατάτμηση χρησιμοποιώντας 2 δείκτες για να χωρίσεις το S στα L και E U G (παρόμοια μέθοδος μπορεί να χωρίσει το E U G σε E και G).

3 2 5 1 0 7 3 5 9 2 7 9 8 9 7 6 9 (οδηγός = 6)

- ◆ Επανάλαβε μέχρι την αντιμετάθεση των j και k :
  - Σάρωσε το j προς τα δεξιά μέχρι να βρεθεί στοιχείο  $\geq x$ .
  - Σάρωσε k προς τα αριστερά μέχρι να βρεθεί στοιχείο  $< x$ .
  - Αντάλλαξε τα στοιχεία στις θέσεις με δείκτες j και k



# Σύνοψη Αλγορίθμων Ταξινόμησης

Αλγόριθμος	Χρόνος	Σημειώσεις
Επιλεκτική ταξινόμηση	$O(n^2)$	<ul style="list-style-type: none"><li>◆ in-place</li><li>◆ αργός (καλός για μικρές εισόδους)</li></ul>
Ενθετική ταξινόμηση	$O(n^2)$	<ul style="list-style-type: none"><li>◆ in-place</li><li>◆ αργός (καλός για μικρές εισόδους)</li></ul>
Ταχυταξινόμηση	$O(n \log n)$ expected	<ul style="list-style-type: none"><li>◆ in-place, συνθήκες τυχειότητας</li><li>◆ γρήγορος (καλός για μεγάλες εισόδους)</li></ul>
Ταξινόμηση σωρού	$O(n \log n)$	<ul style="list-style-type: none"><li>◆ in-place</li><li>◆ γρήγορος (καλός για μεγάλες εισόδους)</li></ul>
Συνχωνευτική		<ul style="list-style-type: none"><li>◆ Ακολουθιακή πρόσβαση δεδομένων</li></ul>

# Τέλος Ενότητας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

# Σημείωμα αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

- Ως Μη Εμπορική ορίζεται η χρήση:
  - που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
  - που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
  - που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο
- Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Ιωάννης Τόλλης 2015. «Αλγόριθμοι και πολυπλοκότητα. Ταχυταξινόμηση (Quick-Sort)». Έκδοση: 1.0. Ηράκλειο 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<https://opencourses.uoc.gr/courses/course/view.php?id=368>



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.