



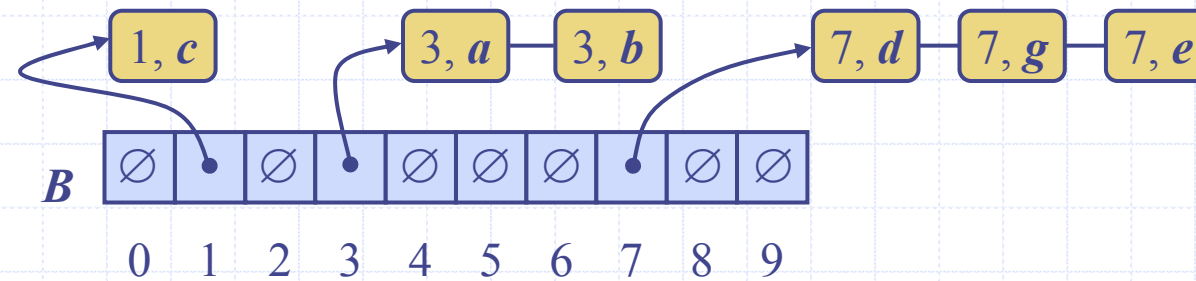
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Αλγόριθμοι και πολυπλοκότητα

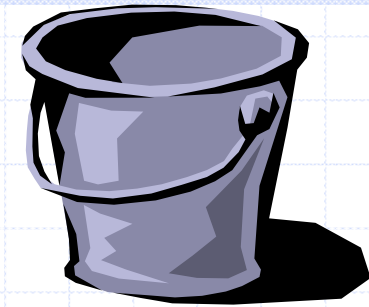
## Bucket-Sort και Radix-Sort

Ιωάννης Τόλλης  
Τμήμα Επιστήμης Υπολογιστών

# Bucket-Sort και Radix-Sort



# Bucket-Sort



◆ Έστω  $S$  μια ακολουθία (key, element) αντικειμένων με κλειδιά στο διάστημα  $[0, N-1]$

◆ Η Bucket-sort χρησιμοποιεί τα κλειδιά σαν δείκτες σε έναν βοηθητικό πίνακα  $B$  από ακολουθίες (buckets)

Φάση 1: Άδειασμα ακολουθιών  $S$  μετακινώντας κάθε αντικείμενο  $(k, o)$  μέσα στο bucket του  $B[k]$

Φάση 2: Για  $i = 0, \dots, N-1$ , μετακίνησε τα αντικείμενα του bucket  $B[i]$  στο τέλος της ακολουθίας  $S$

◆ Ανάλυση:

- Η φάση 1 παίρνει χρόνο  $O(n)$
- Η φάση 2 παίρνει χρόνο  $O(n + N)$
- Η Bucket-sort παίρνει χρόνο

$O(n + N)$

**Algorithm** *bucketSort*( $S, N$ )

**Input** sequence  $S$  of (key, element) items with keys in the range  $[0, N-1]$

**Output** sequence  $S$  sorted by increasing keys

$B \leftarrow$  array of  $N$  empty sequences

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

$f \leftarrow S.first()$

$(k, o) \leftarrow S.remove(f)$

$B[k].insertLast((k, o))$

**for**  $i \leftarrow 0$  **to**  $N-1$

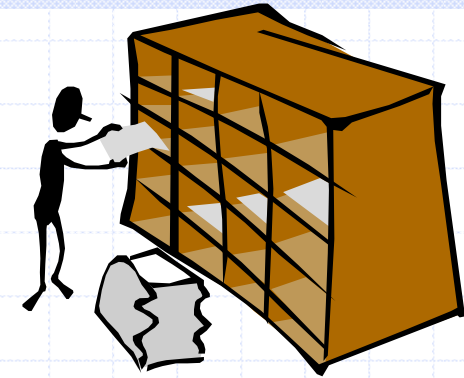
**while**  $\neg B[i].isEmpty()$

$f \leftarrow B[i].first()$

$(k, o) \leftarrow B[i].remove(f)$

$S.insertLast((k, o))$

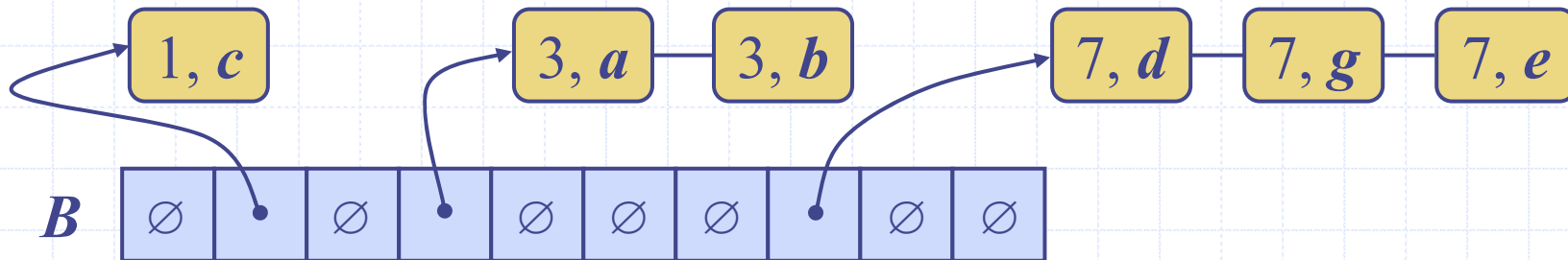
# Παράδειγμα



◆ Εύρος κλειδιών  $[0, 9]$



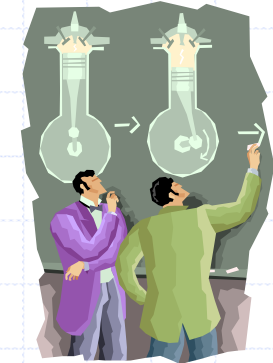
↓ Φάση 1



↓ Φάση 2



# Ιδιότητες και Επεκτάσεις



## ◆ Key-type Ιδιότητα

- Τα κλειδιά χρησιμοποιούνται σαν δείκτες σε πίνακα και δεν μπορούν να είναι αυθαίρετα αντικείμενα
- Δεν υπάρχει εξωτερικός συγκριτής

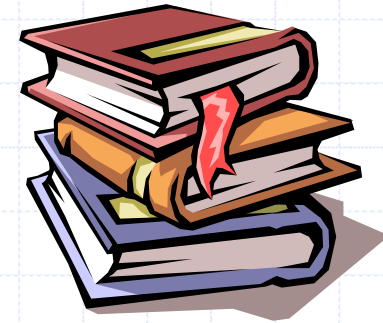
## ◆ **Stable** Sort ιδιότητα

- Η σχετική σειρά οποιωνδήποτε δύο αντικειμένων με το ίδιο κλειδί διατηρείται μετά την εκτέλεση του αλγορίθμου

## Επεκτάσεις

- Ακέραια κλειδιά στο διάστημα  $[a, b]$ 
  - ◆ Τοποθέτηση αντικειμένου  $(k, o)$  στο bucket  $B[k - a]$
- Τα κλειδιά συμβολοσειρών είναι από ένα σύνολο  $D$  πιθανών συμβολοσειρών, όπου  $D$  έχει σταθερό μέγεθος (π.χ., τα ονόματα των 50 πολιτειών των Η.Π.Α.)
  - ◆ Ταξινόμησε το  $D$  και υπολόγισε την τάξη  $r(k)$  της κάθε συμβολοσειράς  $k$  του  $D$  στην ταξινομημένη ακολουθία
  - ◆ Τοποθέτησε το αντικείμενο  $(k, o)$  μέσα στο bucket  $B[r(k)]$

# Λεξικογραφική διάταξη



- ◆ Ένα  $d$ -tuple είναι μια ακολουθία από  $d$  κλειδιά  $(k_1, k_2, \dots, k_d)$ , όπου το κλειδί  $k_i$  είναι η  $i$ -τη διάσταση του tuple
- ◆ Παράδειγμα:
  - Οι καρτεσιανές συντεταγμένες ενός σημείου είναι ένα 3-tuple
- ◆ Η λεξικογραφική διάταξη δύο  $d$ -tuples ορίζεται αναδρομικά ως εξής

$$(x_1, x_2, \dots, x_d) < (y_1, y_2, \dots, y_d)$$



$$x_1 < y_1 \vee x_1 = y_1 \wedge (x_2, \dots, x_d) < (y_2, \dots, y_d)$$

Π.χ., τα tuples συγκρίνονται πρώτα ως προς την πρώτη διάσταση, μετά ως προς την δεύτερη, κ.οκ.

# Λεξικογραφική ταξινόμηση

- ◆ Έστω  $C_i$  ένας συγκριτής ο οποίος συγκρίνει δύο tuples ως προς την  $i$ -τη διάστασή τους
- ◆ Έστω  $stableSort(S, C)$  είναι ένας σταθερός αλγόριθμος ταξινόμησης που χρησιμοποιεί τον συγκριτή  $C$
- ◆ Η λεξικογραφική ταξινόμηση ταξινομεί μια ακολουθία  $d$ -tuples με λεξικογραφική σειρά εκτελώντας  $d$  φορές τον  $stableSort$ , μία φορά για κάθε διάσταση
- ◆ Η λεξικογραφική ταξινόμηση εκτελείται σε χρόνο  $O(dT(n))$ , όπου  $T(n)$  είναι ο χρόνος εκτέλεσης του  $stableSort$

**Algorithm** *lexicographicSort(S)*

**Input** sequence  $S$  of  $d$ -tuples

**Output** sequence  $S$  sorted in lexicographic order

```
for  $i \leftarrow d$  downto 1  
     $stableSort(S, C_i)$ 
```

Παράδειγμα:

(7,4,6) (5,1,5) (2,4,6) (2, 1, 4) (3, 2, 4)

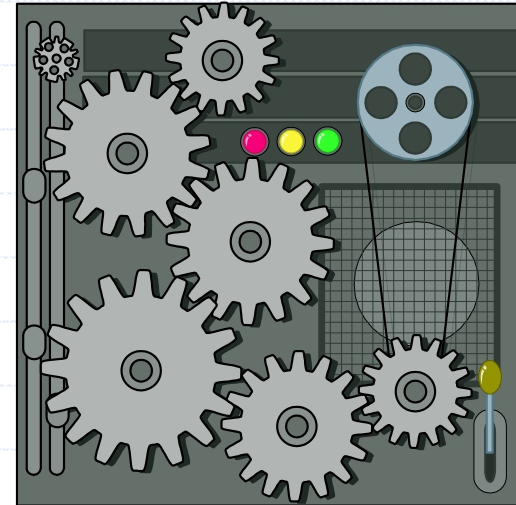
(2, 1, 4) (3, 2, 4) (5,1,5) (7,4,6) (2,4,6)

(2, 1, 4) (5,1,5) (3, 2, 4) (7,4,6) (2,4,6)

(2, 1, 4) (2,4,6) (3, 2, 4) (5,1,5) (7,4,6)

# Radix-Sort (§ 4.5.2)

- ◆ Η Radix-sort είναι μια άλλη έκδοση της λεξικογραφικής ταξινόμησης που χρησιμοποιεί bucket-sort σαν το σταθερό αλγοριθμό σε κάθε διάσταση
- ◆ Η Radix-sort είναι εφαρμόσιμη σε tuples όπου τα κλειδιά σε κάθε διάσταση  $i$  είναι ακέραιοι στο διάστημα  $[0, N - 1]$
- ◆ Η Radix-sort εκτελείται σε χρόνο  $O(d(n + N))$



## Algorithm *radixSort(S, N)*

**Input** sequence  $S$  of  $d$ -tuples such that  $(0, \dots, 0) \leq (x_1, \dots, x_d)$  and  $(x_1, \dots, x_d) \leq (N - 1, \dots, N - 1)$  for each tuple  $(x_1, \dots, x_d)$  in  $S$

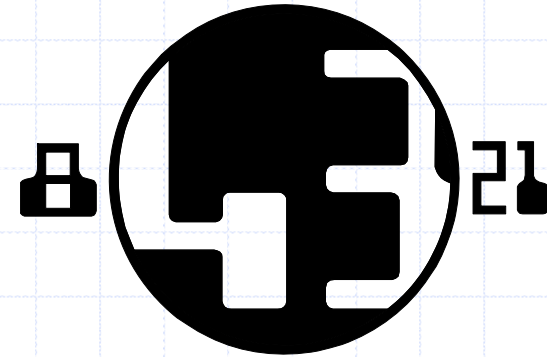
**Output** sequence  $S$  sorted in lexicographic order

**for**  $i \leftarrow d$  **downto** 1

*bucketSort(S, N)*



# Radix-Sort για δυναδικούς αριθμούς



- ◆ Θεωρείστε μια ακολουθία από  $n$   $b$ -bit ακεραίους

$$x = x_{b-1} \dots x_1 x_0$$

- ◆ Αναπαριστούμε κάθε στοιχείο σαν ένα  $b$ -tuple από ακεραίους στο διάστημα  $[0, 1]$  και εφαρμόζουμε radix-sort με  $N = 2$

- ◆ Αυτή η εφαρμογή του radix-sort αλγορίθμου εκτελείται σε  $O(bn)$  χρόνο

- ◆ Για παράδειγμα, μπορούμε να ταξινομήσουμε μια ακολουθία από 32-bit ακεραίους σε γραμμικό χρόνο

**Algorithm** *binaryRadixSort(S)*

**Input** sequence  $S$  of  $b$ -bit integers

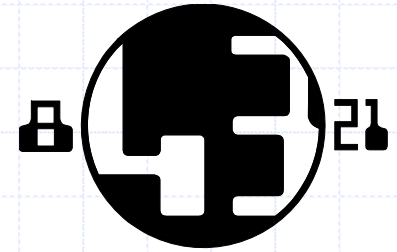
**Output** sequence  $S$  sorted  
replace each element  $x$  of  $S$  with the item  $(0, x)$

**for**  $i \leftarrow 0$  **to**  $b - 1$

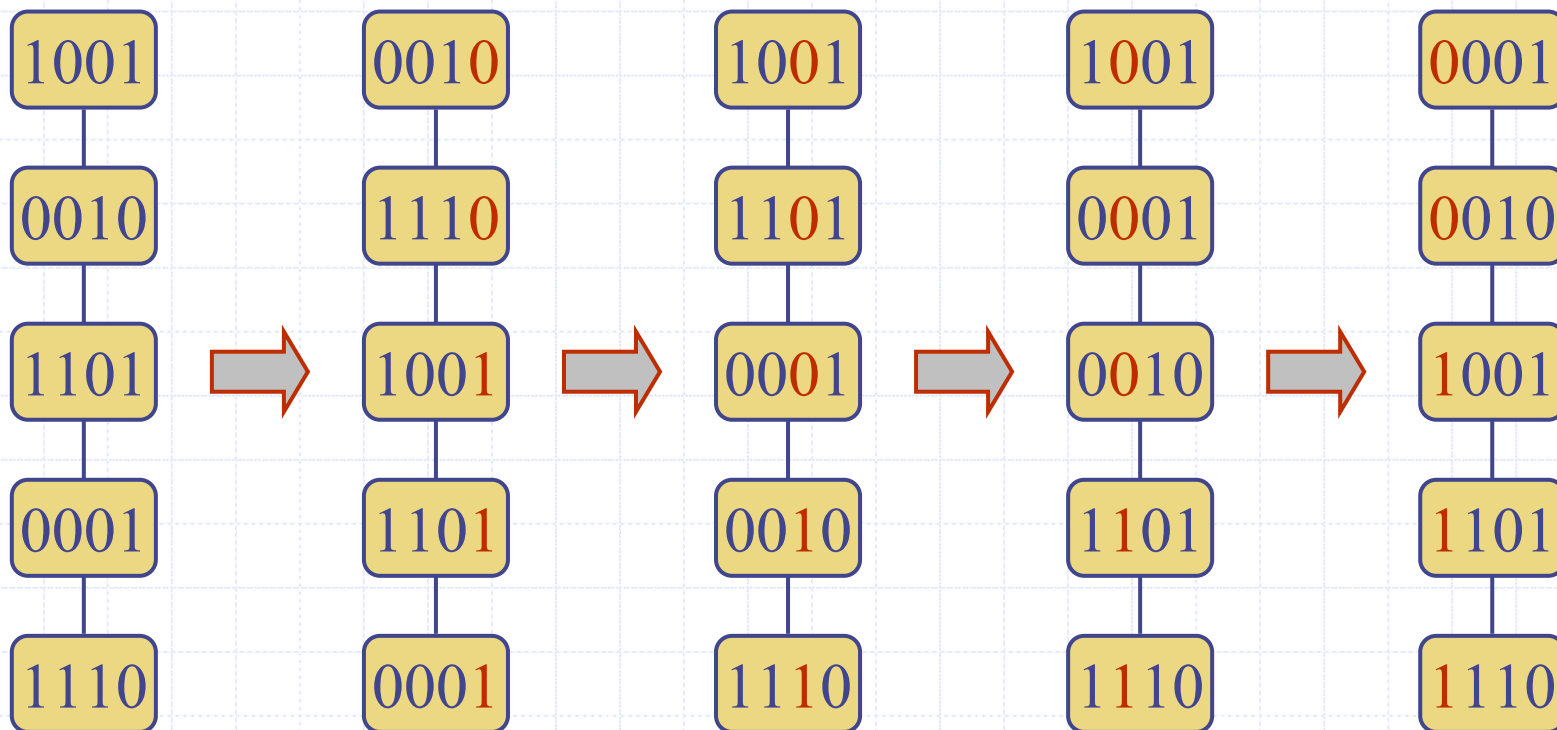
replace the key  $k$  of each item  $(k, x)$  of  $S$  with bit  $x_i$  of  $x$

*bucketSort(S, 2)*

# Παράδειγμα



- ◆ Ταξινόμηση μιας ακολουθίας ακεραίων 4-bit (τετράμπιτων)



# 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. Bucket-Sort και Radix-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.