

Οι διαφάνειες βασίζονται σε αυτές του
ακόλουθου μαθήματος:

Introduction to Algorithms (6-046J), MIT

[https://ocw.mit.edu/courses/electrical-
engineering-and-computer-science/6-046j-
introduction-to-algorithms-sma-5503-fall-2005/](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046j-introduction-to-algorithms-sma-5503-fall-2005/)

Οι διαφάνειες του ανωτέρω μαθήματος
δίνονται υπό την άδεια «Creative Commons
Attribution-NonCommercial-ShareAlike 3.0»

Πόσο γρήγορα μπορούμε να ταξινομήσουμε;

Όλοι οι αλγόριθμοι ταξινόμησης που έχουμε δει είναι αλγόριθμοι **συγκριτικής ταξινόμησης**: χρησιμοποιούν μόνο συγκρίσεις για να καθορίζουν τη σχετική διάταξη των στοιχείων.

- Π.χ., ενθετική ταξινόμηση, συγχωνευτική ταξινόμηση, ταχυταξινόμηση, ταξινόμηση σωρού.

Ο καλύτερος χρόνος χειρότερης περίπτωσης που έχουμε δει για συγκριτική ταξινόμηση είναι $O(n \lg n)$.

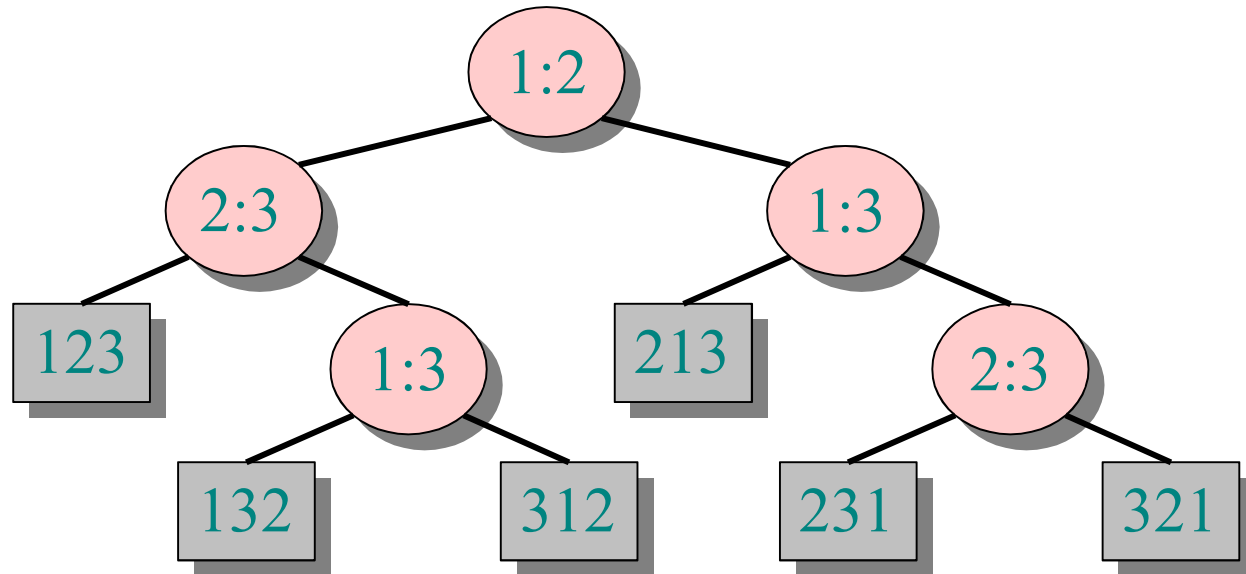
Είναι ο χρόνος $O(n \lg n)$ το καλύτερο που μπορούμε να κάνουμε;

Τα δένδρα μπορούν να βοηθήσουν στην απάντηση αυτής της ερώτησης.

Παράδειγμα Δένδρου Απόφασης

Ταξινόμηση της

$\langle a_1, a_2, \dots, a_n \rangle$



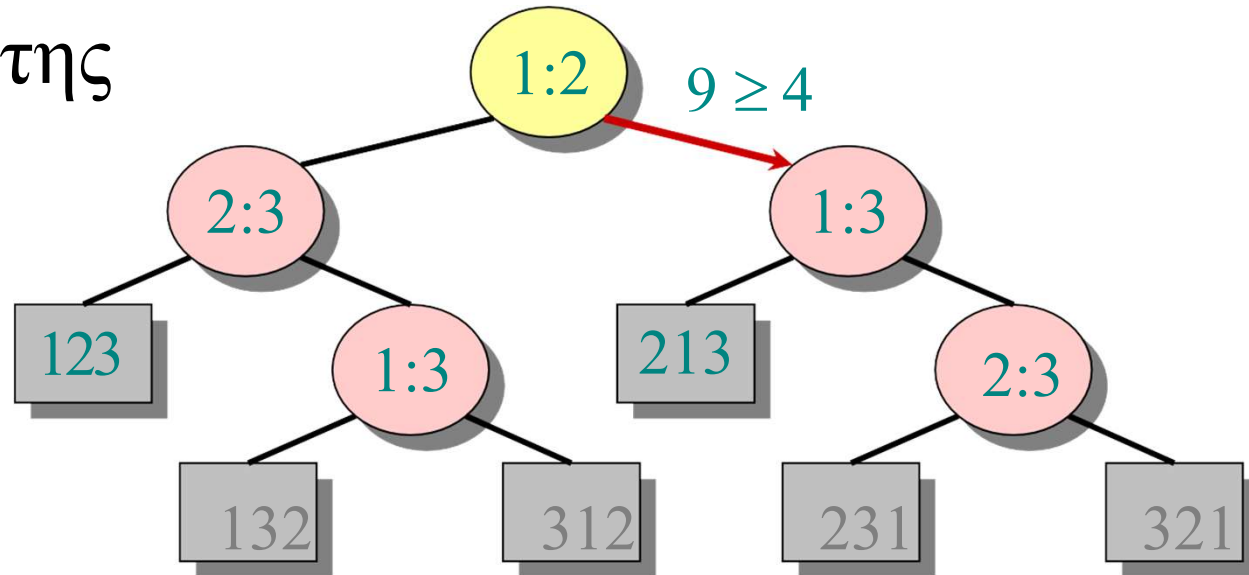
Κάθε εσωτερικός κόμβος επιγράφεται με $i:j$ για $i, j \in \{1, 2, \dots, n\}$.

- Το αριστερό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \leq a_j$.
- Το δεξιό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \geq a_j$.

Παράδειγμα Δένδρου Απόφασης

Ταξινόμηση της

$$\langle a_1, a_2, a_3 \rangle \\ = \langle 9, 4, 6 \rangle$$

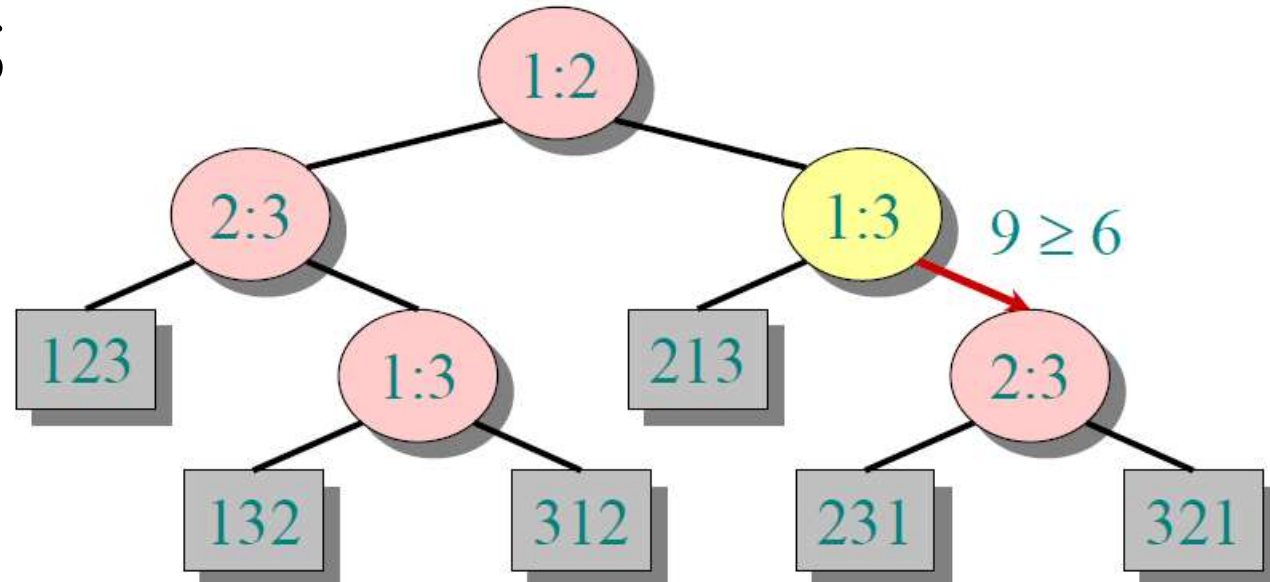


Κάθε εσωτερικός κόμβος επιγράφεται με $i:j$ για $i, j \in \{1, 2, \dots, n\}$.

- Το αριστερό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \leq a_j$.
- Το δεξιό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \geq a_j$.

Παράδειγμα Δένδρου Απόφασης

Ταξινόμηση της
 $\langle a_1, a_2, a_3 \rangle$
 $= \langle 9, 4, 6 \rangle$



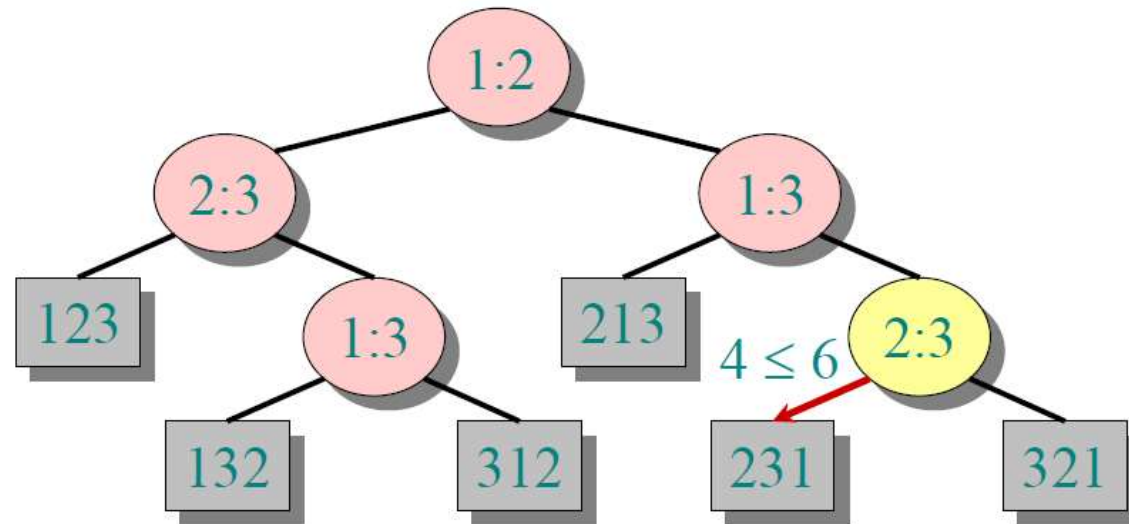
Κάθε εσωτερικός κόμβος επιγράφεται με $i:j$ για $i, j \in \{1, 2, \dots, n\}$.

- Το αριστερό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \leq a_j$.
- Το δεξιό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \geq a_j$.

Παράδειγμα Δένδρου Απόφασης

Ταξινόμηση της

$$\langle a_1, a_2, a_3 \rangle \\ = \langle 9, 4, 6 \rangle$$



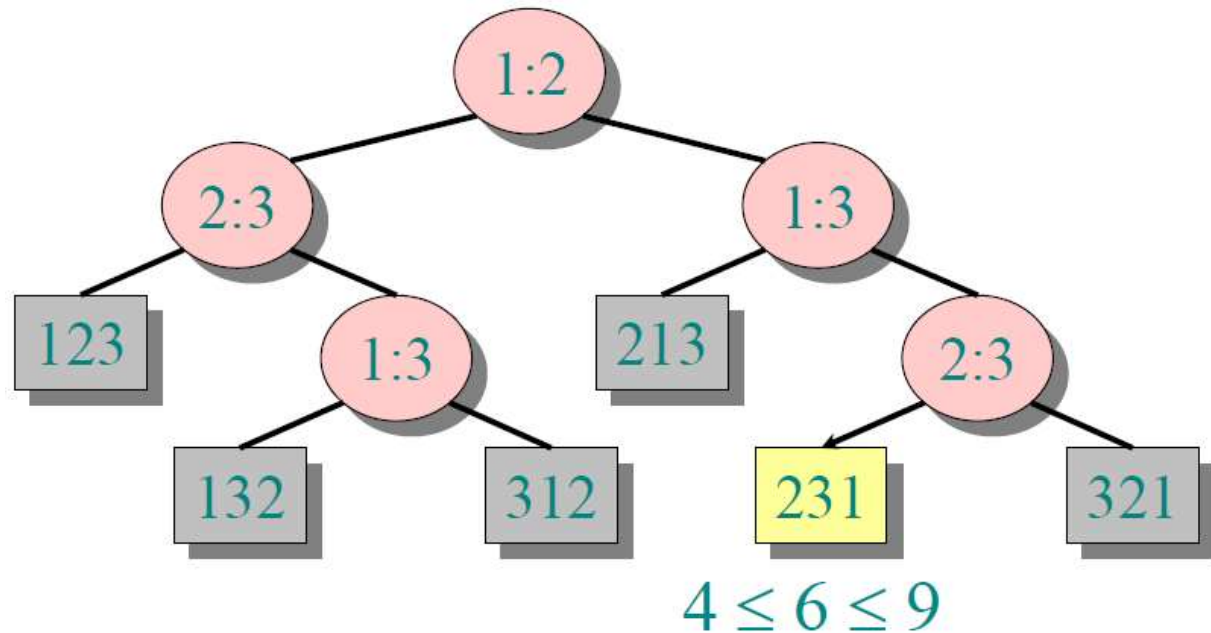
Κάθε εσωτερικός κόμβος επιγράφεται με $i:j$ για $i, j \in \{1, 2, \dots, n\}$.

- Το αριστερό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \leq a_j$.
- Το δεξιό υποδένδρο δείχνει τις συγκρίσεις που έπονται αν $a_i \geq a_j$.

Παράδειγμα Δένδρου Απόφασης

Ταξινόμηση της

$$\langle a_1, a_2, a_3 \rangle \\ = \langle 9, 4, 6 \rangle$$



Κάθε φύλλο περιέχει μία μετάθεση $\langle \pi(1), \pi(2), \dots, \pi(n) \rangle$ η οποία δείχνει ότι η διάταξη $a_{\pi(1)} \leq a_{\pi(2)} \leq \dots \leq a_{\pi(n)}$ έχει επιτευχθεί.

Μοντέλο Δένδρου Απόφασης

Ένα δένδρο απόφασης μπορεί να αποδώσει την εκτέλεση οποιασδήποτε συγκριτικής ταξινόμησης:

- Ένα δένδρο για κάθε μέγεθος εισόδου n .
- Δες τον αλγόριθμο ως διαχωριστή, οποτεδήποτε αυτός συγκρίνει δύο στοιχεία.
- Το δένδρο περιέχει τις συγκρίσεις για όλες τις πιθανές ροές ελέγχου.
- Ο χρόνος εκτέλεσης του αλγορίθμου = το μήκος του μονοπατιού που ακολουθείται.
- Χρόνος εκτέλεσης χειρότερης περίπτωσης = ύψος του δένδρου.

Κάτω όριο για Ταξινόμηση με βάση το Δένδρο Απόφασης

Θεώρημα. Κάθε δέντρο απόφασης που μπορεί να ταξινομήσει n στοιχεία πρέπει να έχει ύψος $\Omega(n \lg n)$.

Απόδειξη. Το δένδρο πρέπει να περιέχει $\geq n!$ φύλλα, αφού υπάρχουν $n!$ πιθανές μεταθέσεις. Ένα δυαδικό δένδρο ύψους h έχει $\leq 2^h$ φύλλα. Έτσι, $n! \leq 2^h$.

$$\begin{aligned} \therefore h &\geq \lg(n!) \\ &\geq \lg((n/e)^n) \\ &= n \lg n - n \lg e \\ &= \Omega(n \lg n) \end{aligned}$$

(\lg είναι μονότονη αύξουσα συνάρτηση)
(προσέγγιση Stirling)

Κάτω όριο για Ταξινόμηση με βάση το Δένδρο Απόφασης

Πόρισμα. Η ταξινόμηση σωρού και η συγχωνευτική ταξινόμηση είναι ασυμπτωτικά βέλτιστοι αλγόριθμοι συγκριτικής ταξινόμησης.

Ταξινόμηση σε γραμμικό χρόνο

Απαριθμητική ταξινόμηση: Όχι συγκρίσεις μεταξύ των στοιχείων.

- **Είσοδος:** $A[1 \dots n]$, όπου $A[j] \in \{1, 2, \dots, k\}$.
- **Έξοδος:** $B[1 \dots n]$, ταξινομημένος.
- **Βοηθητικός πίνακας:** $C[1 \dots k]$.

Απαριθμητική ταξινόμηση

for $i \leftarrow 1$ **to** k

do $C[i] \leftarrow 0$

for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$

for $i \leftarrow 2$ **to** k

do $C[i] \leftarrow C[i] + C[i-1]$

for $j \leftarrow n$ **downto** 1

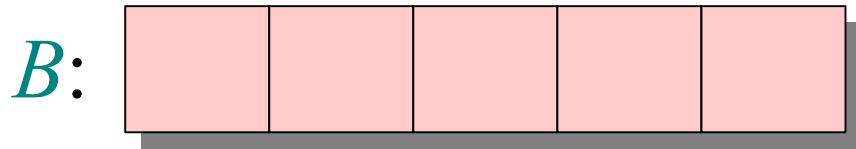
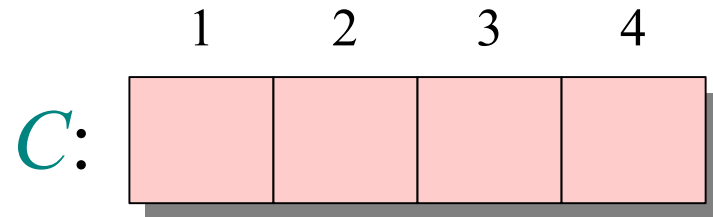
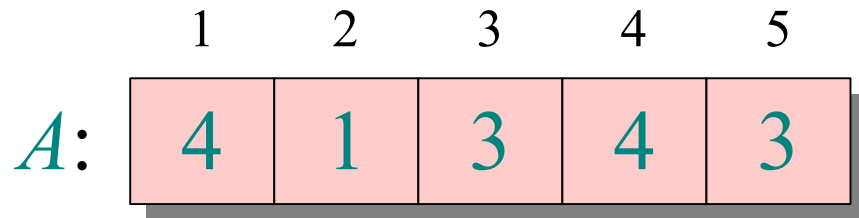
do $B[C[A[j]]] \leftarrow A[j]$

$C[A[j]] \leftarrow C[A[j]] - 1$

▷ $C[i] = |\{\text{key} = i\}|$

▷ $C[i] = |\{\text{key} \leq i\}|$

Παράδειγμα Απαριθμητικής Ταξινόμησης



Βρόχος 1

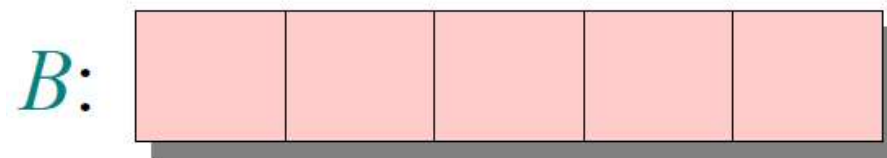
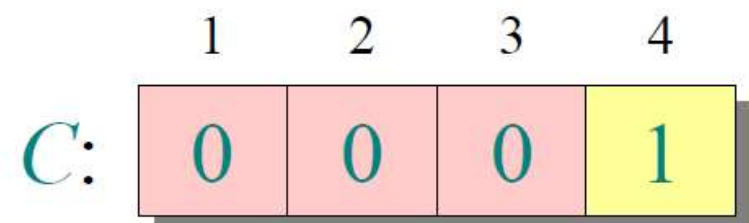
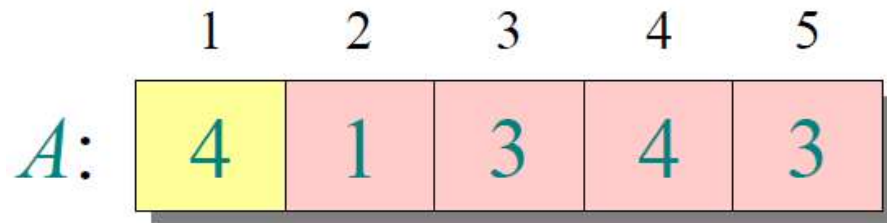
	1	2	3	4	5
<i>A</i> :	4	1	3	4	3

	1	2	3	4
<i>C</i> :	0	0	0	0

<i>B</i> :					
------------	--	--	--	--	--

```
for  $i \leftarrow 1$  to  $k$   
  do  $C[i] \leftarrow 0$ 
```

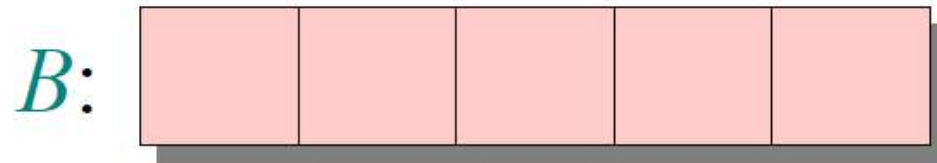
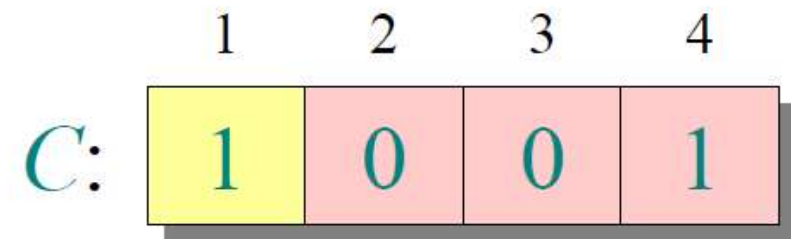
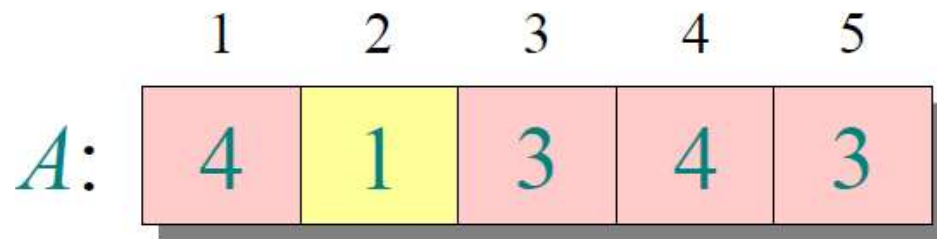
Βρόχος 2



for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$ $\triangleright C[i] = |\{\text{key} = i\}|$

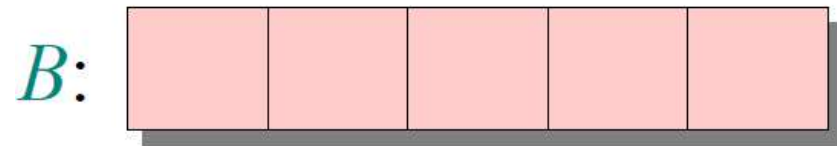
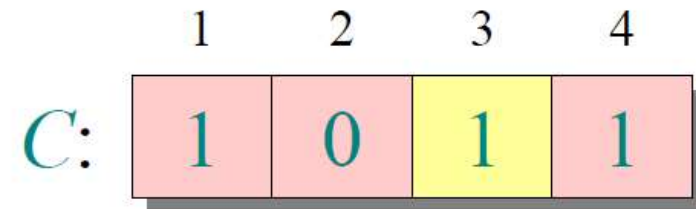
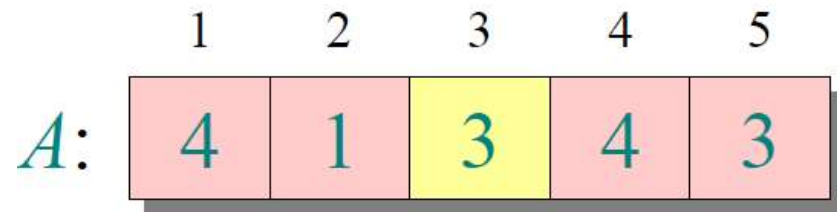
Βρόχος 2



for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$ $\triangleright C[i] = |\{\text{key} = i\}|$

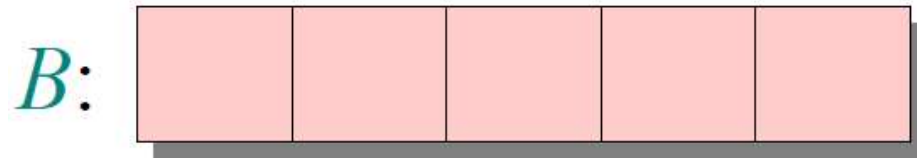
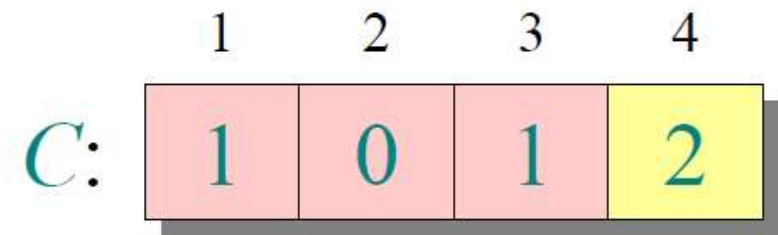
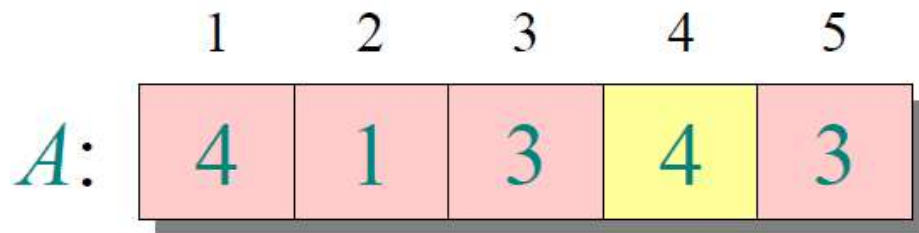
Βρόχος 2



for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$ $\triangleright C[i] = |\{\text{key} = i\}|$

Βρόχος 2

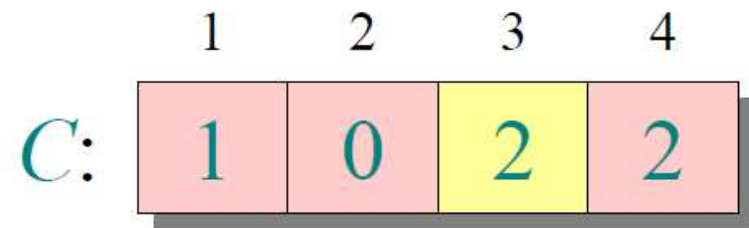
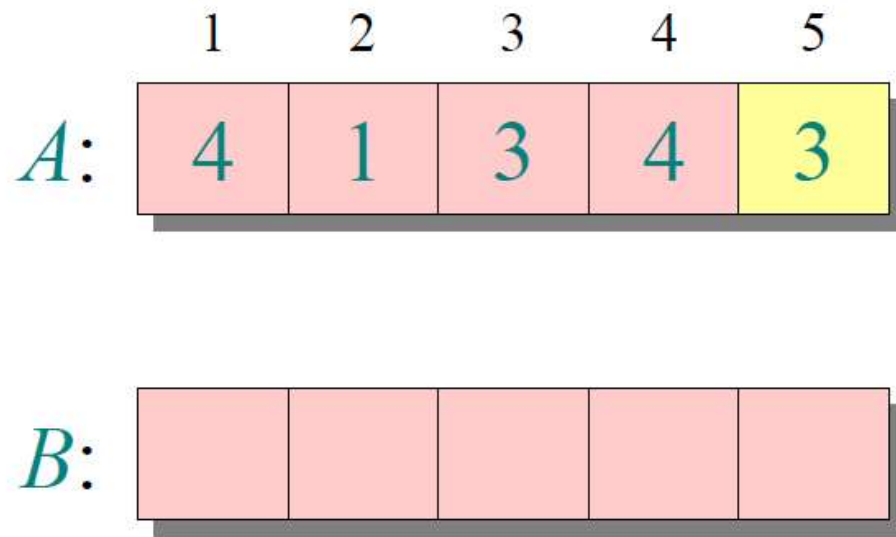


for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$

$\triangleright C[i] = |\{\text{key} = i\}|$

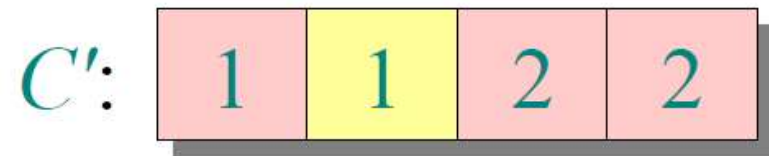
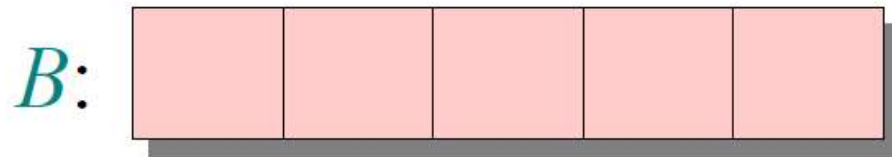
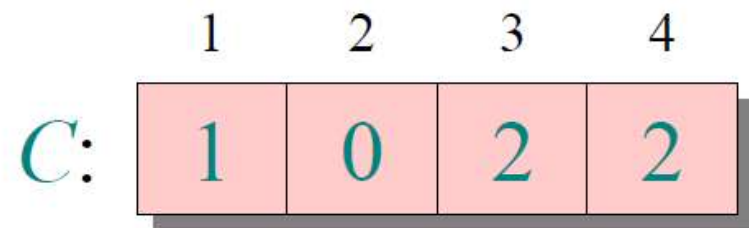
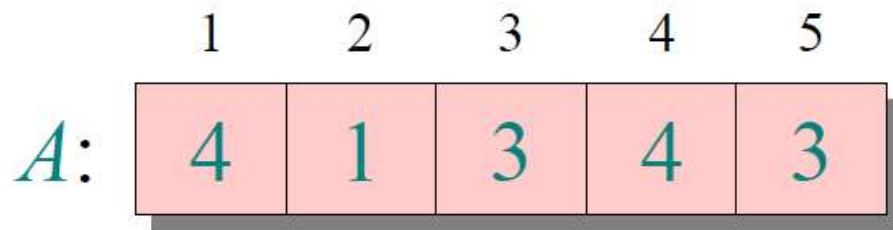
Βρόχος 2



for $j \leftarrow 1$ **to** n

do $C[A[j]] \leftarrow C[A[j]] + 1$ $\triangleright C[i] = |\{\text{key} = i\}|$

Βρόχος 3

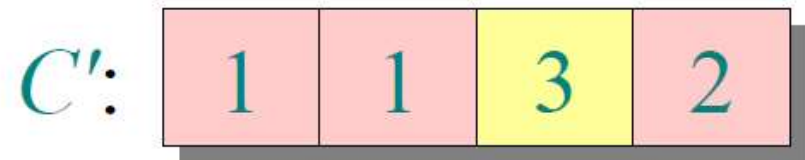
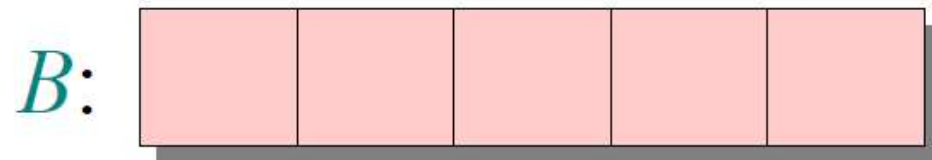
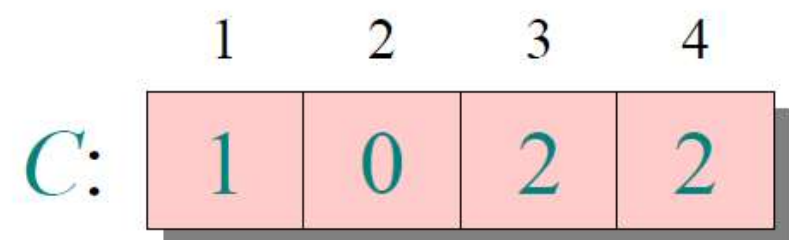
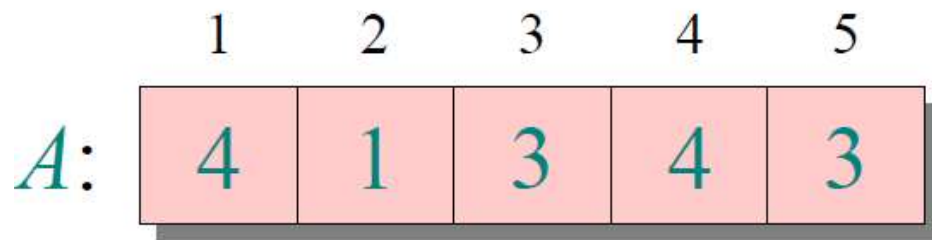


for $i \leftarrow 2$ **to** k

do $C[i] \leftarrow C[i] + C[i-1]$

$\triangleright C[i] = |\{\text{key} \leq i\}|$

Βρόχος 3



for $i \leftarrow 2$ **to** k

do $C[i] \leftarrow C[i] + C[i-1]$

$\triangleright C[i] = |\{\text{key} \leq i\}|$

Βρόχος 3

	1	2	3	4	5
<i>A</i> :	4	1	3	4	3

	1	2	3	4
<i>C</i> :	1	0	2	2

<i>B</i> :					
------------	--	--	--	--	--

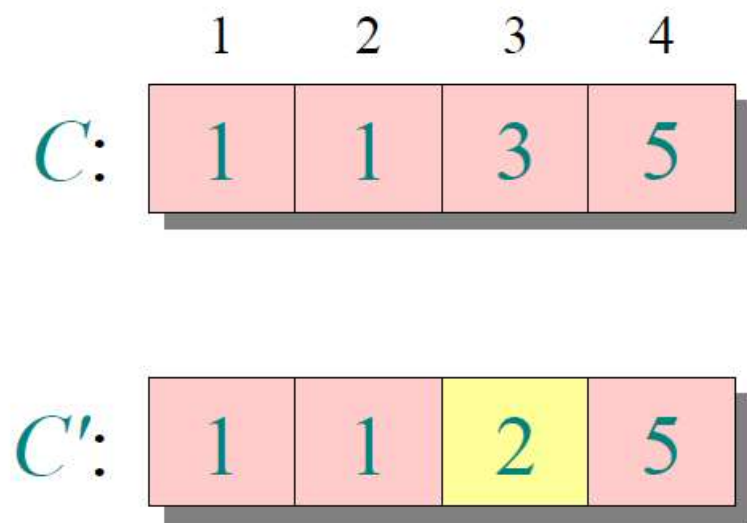
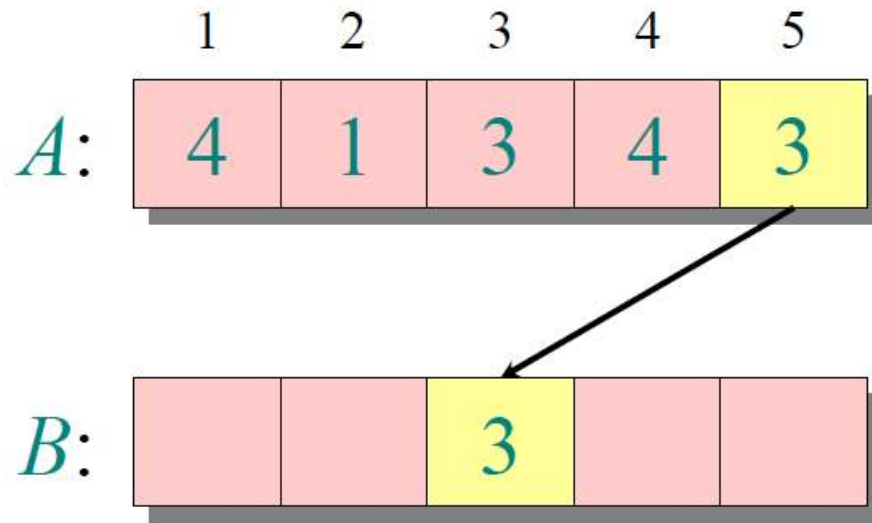
<i>C'</i> :	1	1	3	5
-------------	---	---	---	---

for $i \leftarrow 2$ **to** k

do $C[i] \leftarrow C[i] + C[i-1]$

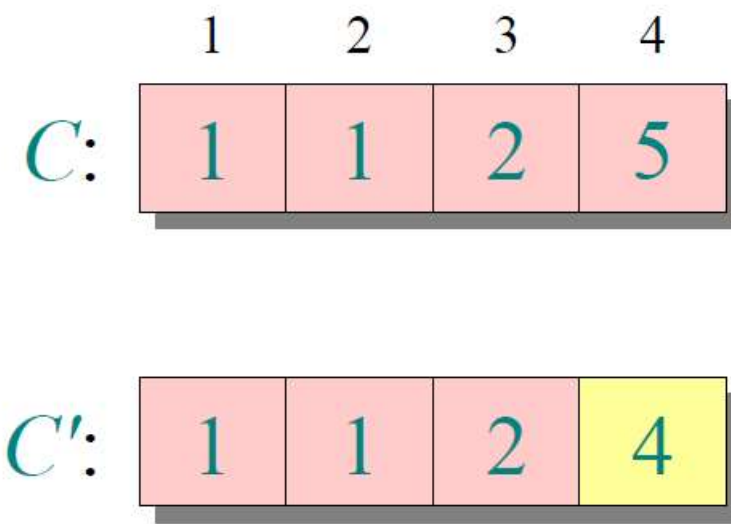
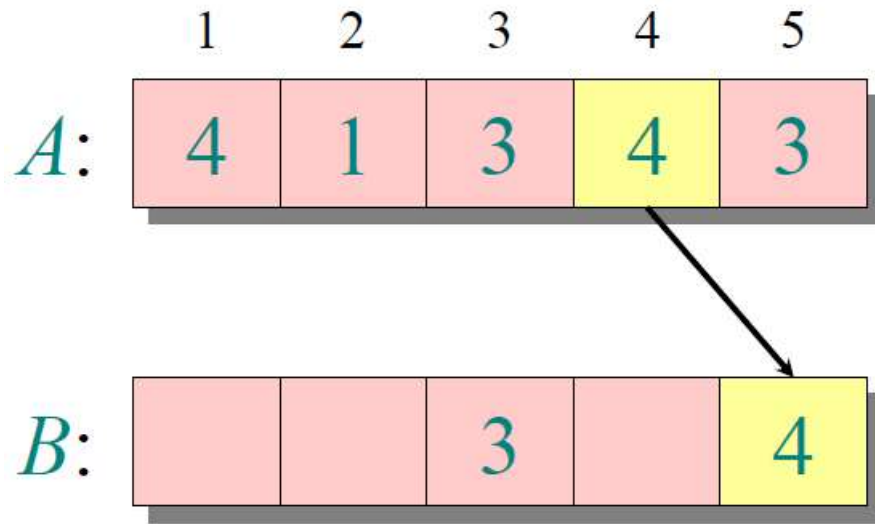
$\triangleright C[i] = |\{\text{key} \leq i\}|$

Βρόχος 4



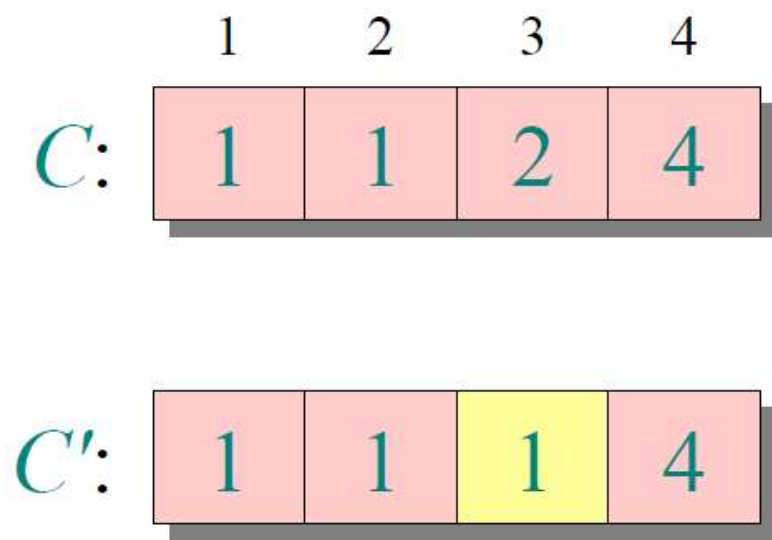
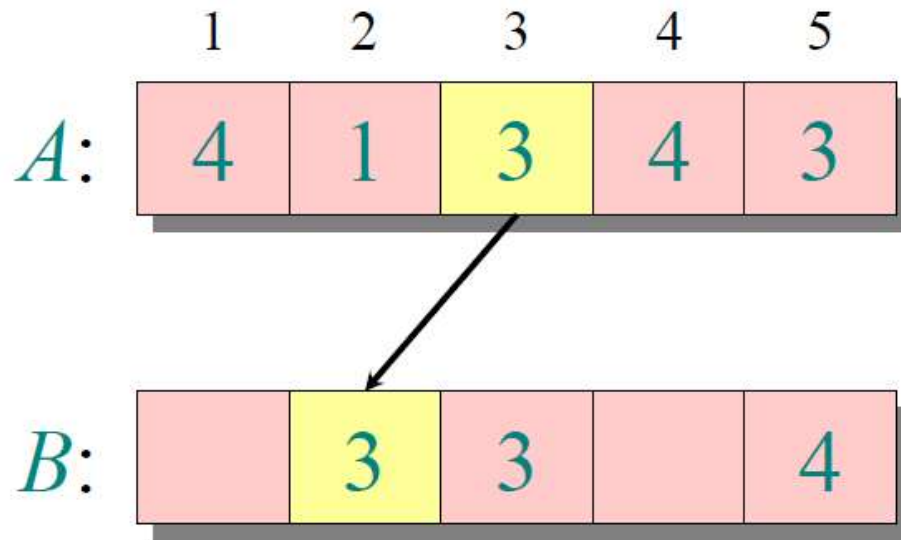
```
for  $j \leftarrow n$  downto 1  
  do  $B[C[A[j]]] \leftarrow A[j]$   
       $C[A[j]] \leftarrow C[A[j]] - 1$ 
```

Βρόχος 4



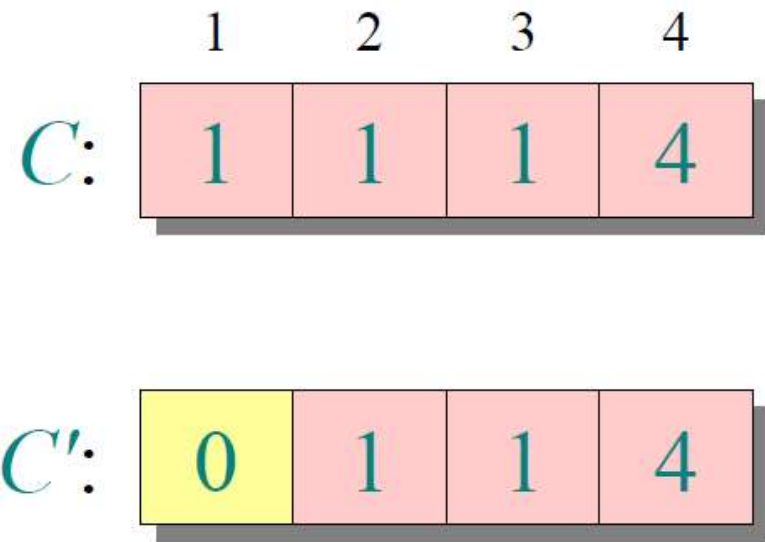
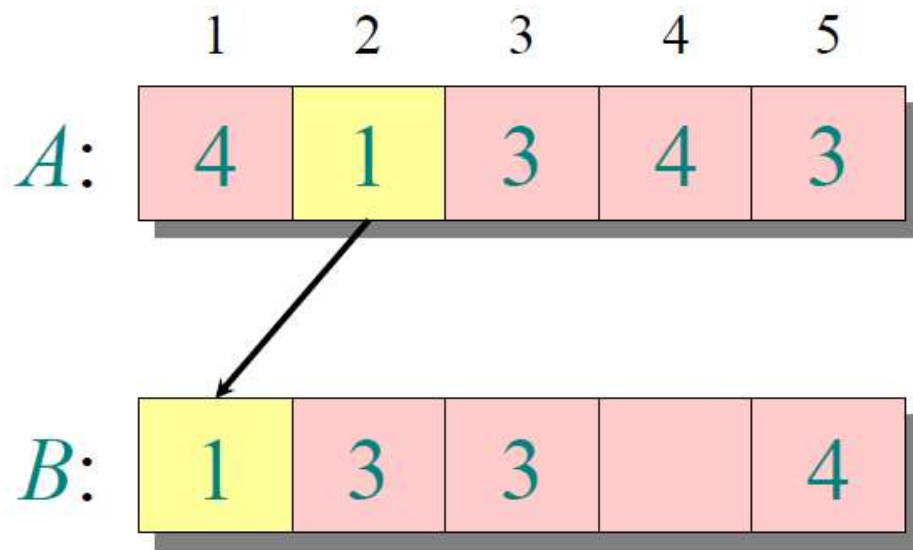
```
for  $j \leftarrow n$  downto 1  
  do  $B[C[A[j]]] \leftarrow A[j]$   
      $C[A[j]] \leftarrow C[A[j]] - 1$ 
```


Βρόχος 4



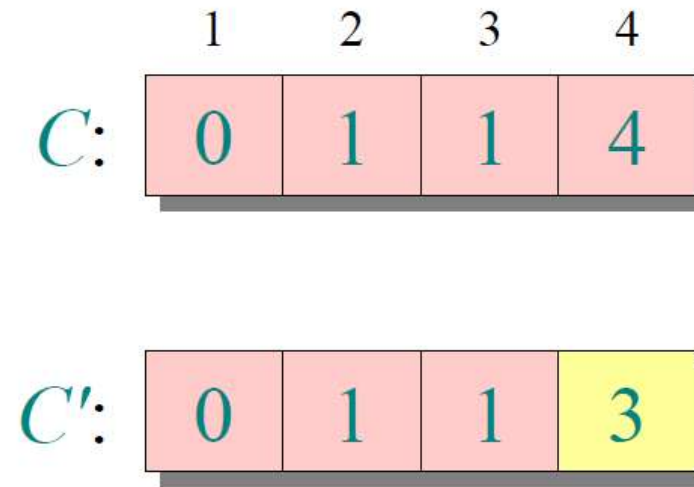
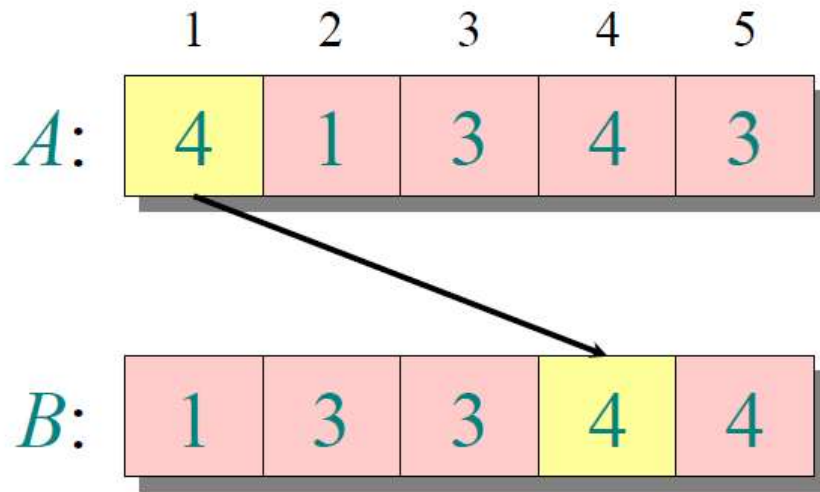
```
for  $j \leftarrow n$  downto 1  
  do  $B[C[A[j]]] \leftarrow A[j]$   
       $C[A[j]] \leftarrow C[A[j]] - 1$ 
```

Βρόχος 4



```
for  $j \leftarrow n$  downto 1  
  do  $B[C[A[j]]] \leftarrow A[j]$   
      $C[A[j]] \leftarrow C[A[j]] - 1$ 
```

Βρόχος 5



```
for  $j \leftarrow n$  downto 1  
  do  $B[C[A[j]]] \leftarrow A[j]$   
      $C[A[j]] \leftarrow C[A[j]] - 1$ 
```

Ανάλυση

$\Theta(k)$ { **for** $i \leftarrow 1$ **to** k
 do $C[i] \leftarrow 0$

$\Theta(n)$ { **for** $j \leftarrow 1$ **to** n
 do $C[A[j]] \leftarrow C[A[j]] + 1$

$\Theta(k)$ { **for** $i \leftarrow 2$ **to** k
 do $C[i] \leftarrow C[i] + C[i-1]$

$\Theta(n)$ { **for** $j \leftarrow n$ **downto** 1
 do $B[C[A[j]]] \leftarrow A[j]$
 $C[A[j]] \leftarrow C[A[j]] - 1$

$\Theta(n + k)$

Χρόνος Εκτέλεσης

Αν $k = O(n)$, τότε η απαριθμητική ταξινόμηση χρειάζεται χρόνο $\Theta(n)$.

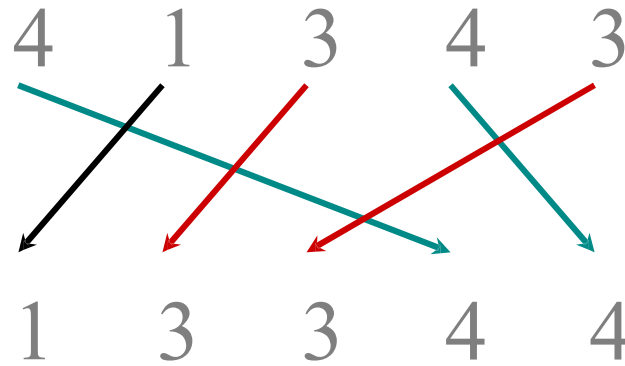
- Αλλά, η ταξινόμηση χρειάζεται $\Omega(n \lg n)$ χρόνο!
- Που είναι η πλάνη;

Απάντηση:

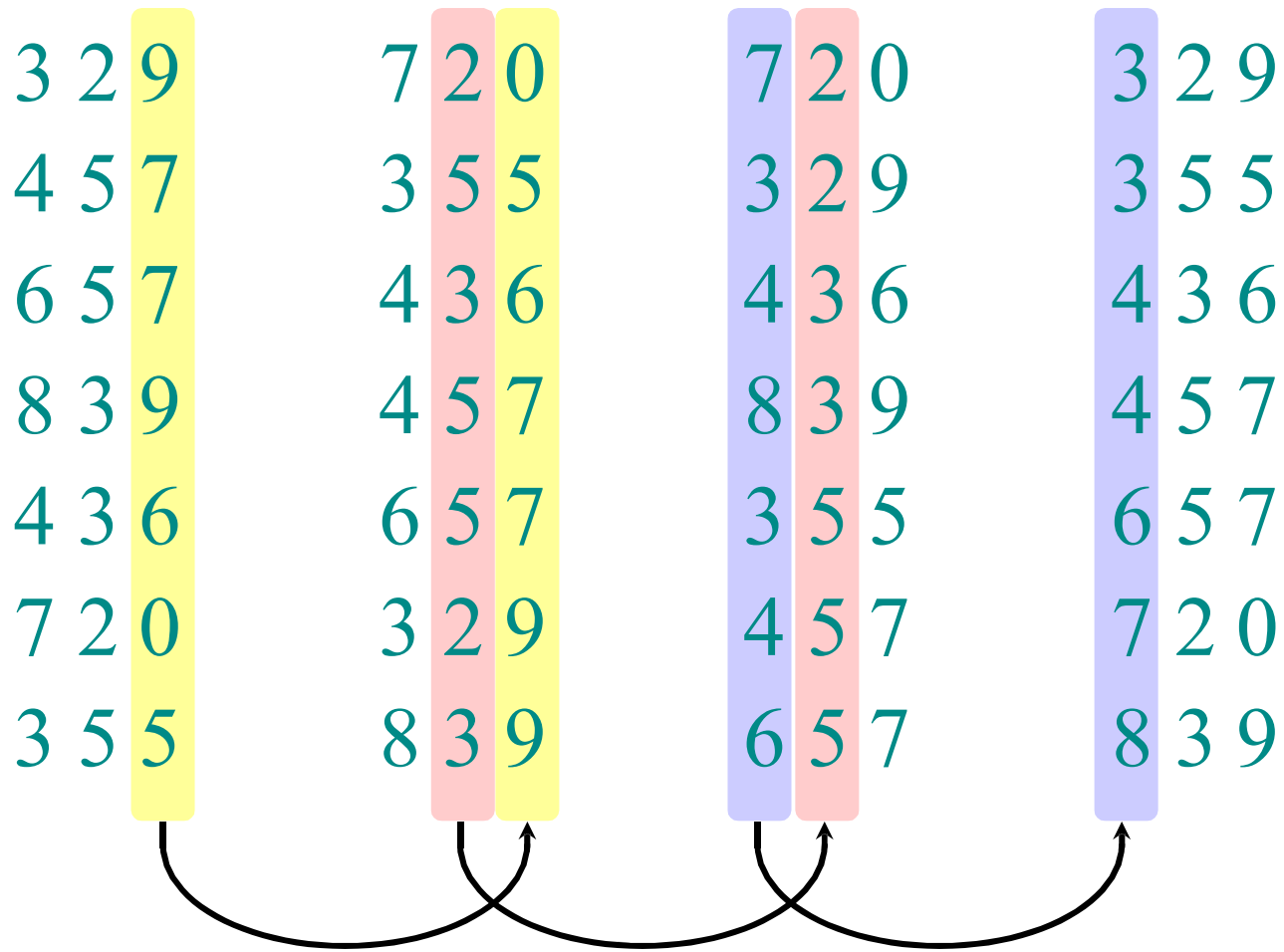
- Η **συγκριτική ταξινόμηση** παίρνει χρόνο $\Omega(n \lg n)$.
- Η απαριθμητική ταξινόμηση δεν είναι **συγκριτική ταξινόμηση**.
- Στην πραγματικότητα, δεν γίνεται καμία σύγκριση μεταξύ των στοιχείων.

Ευσταθής ταξινόμηση

Η απαριθμητική ταξινόμηση είναι ευσταθής: διατηρεί την αρχική διάταξη των ίσων στοιχείων της εισόδου:



Αριθμοτακτική Ταξινόμηση (Radix Sort)

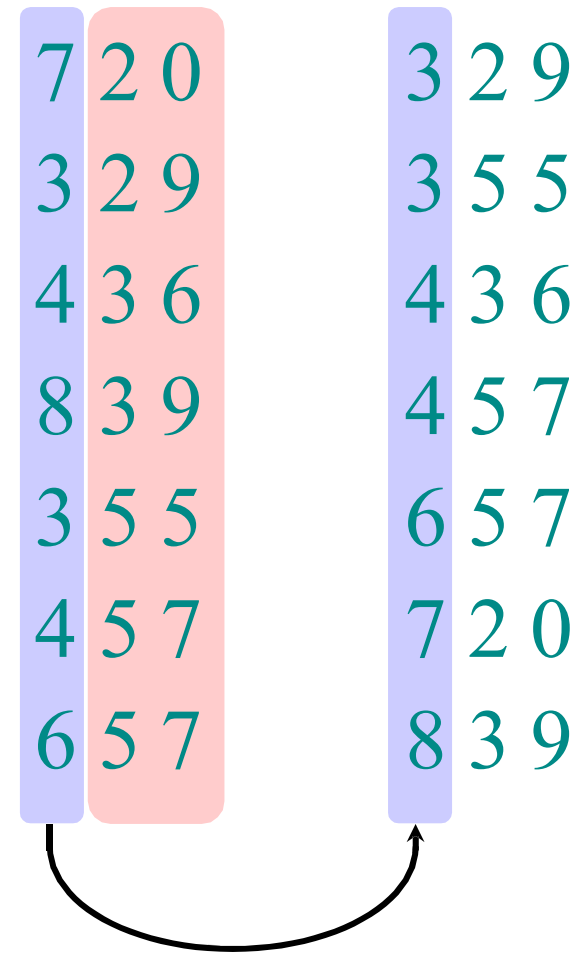


Ορθότητα της αριθμοτακτικής ταξινόμησης

Επαγωγή ως προς τη θέση του ψηφίου

Υποθέτουμε ότι οι αριθμοί είναι ταξινομημένοι στα χαμηλότερης τάξης $t - 1$ ψηφία.

- Ταξινόμηση στο ψηφίο t

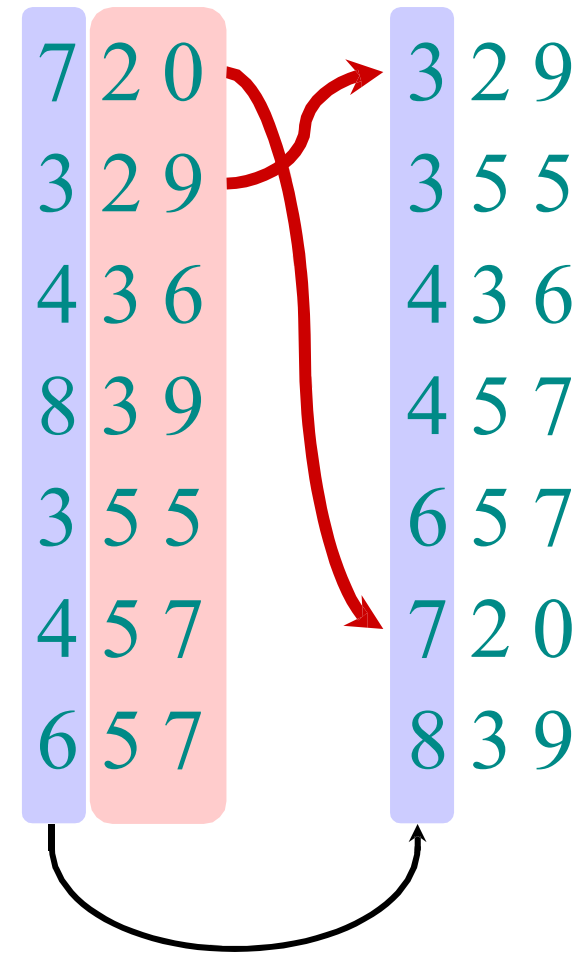


Ορθότητα της αριθμοτακτικής ταξινόμησης

Επαγωγή ως προς τη θέση του ψηφίου

Υποθέτουμε ότι οι αριθμοί είναι ταξινομημένοι στα χαμηλότερης τάξης $t - 1$ ψηφία.

- Ταξινόμηση στο ψηφίο t
 - Δύο αριθμοί οι οποίοι διαφέρουν στο ψηφίο t ταξινομούνται σωστά.

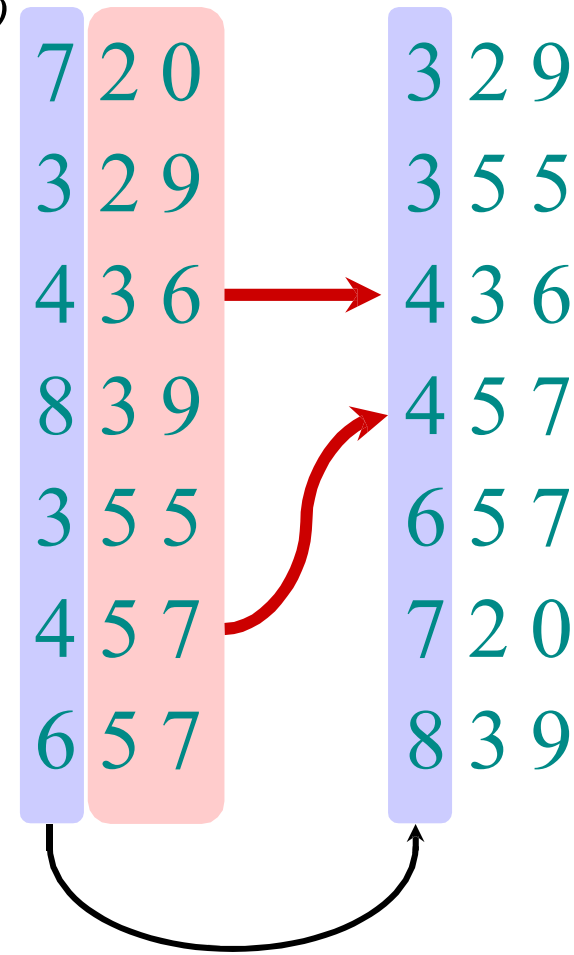


Ορθότητα της αριθμοτακτικής ταξινόμησης

Επαγωγή ως προς τη θέση του ψηφίου

Υποθέτουμε ότι οι αριθμοί είναι ταξινομημένοι στα χαμηλότερης τάξης $t - 1$ ψηφία.

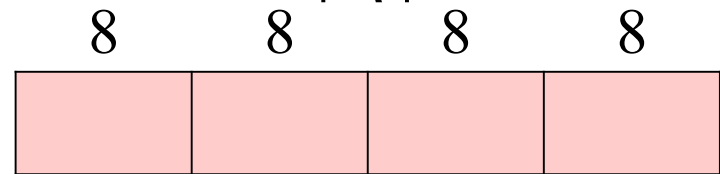
- Ταξινόμηση στο ψηφίο t
 - Δύο αριθμοί οι οποίοι διαφέρουν στο ψηφίο t ταξινομούνται σωστά.
 - Δύο αριθμοί ίδιοι στο ψηφίο t τοποθετούνται στην ίδια σειρά όπως στην είσοδο \Rightarrow σωστή διάταξη.



Ανάλυση της αριθμοτακτικής ταξινόμησης

- Υποθέτουμε απεριθμητική ταξινόμηση σε κάθε πέρασμα.
- Ταξινόμηση n αριθμούς των b δυαδικών ψηφίων ο καθένας.

Παράδειγμα: 32-bit αριθμοί



- Κάθε αριθμός μπορεί να θεωρηθεί σαν να έχει b/r ψηφία βάσης 2^r
 $r = 8 \Rightarrow b/r = 4$ περάσματα της απεριθμητικής ταξινόμησης σε ψηφία βάσης 2^8 ή $r = 16 \Rightarrow b/r = 2$ περάσματα απεριθμητικής ταξινόμησης σε ψηφία βάσης 2^{16} .

Πόσα περάσματα πρέπει να κάνουμε;

Ανάλυση (συν.)

Θυμηθείτε: Η απαριθμητική ταξινόμηση παίρνει χρόνο $\Theta(n + k)$ για να ταξινομήσει n αριθμούς στο εύρος $0 \dots k - 1$.

Αν κάθε b -bit αριθμός σπάσει σε r -bit κομμάτια, κάθε πέρασμα της απαριθμητικής ταξινόμησης παίρνει χρόνο $\Theta(n + 2^r)$. Αφού υπάρχουν b/r περάσματα, έχουμε

$$T(n, b) = \Theta\left(\frac{b}{r}(n + 2^r)\right).$$

Διάλεξε το r για ελαχιστοποίηση του $T(n, b)$:

- Αυξάνοντας το r προκύπτουν λιγότερα περάσματα, αλλά καθώς $r \gg \lg n$, ο χρόνος αυξάνεται εκθετικά.

Επιλογή r

$$T(n, b) = \Theta\left(\frac{b}{r}(n + 2^r)\right).$$

Ελαχιστοποίηση του $T(n, b)$ με παραγωγή και εξίσωση με το 0.

Ή, απλά παρατηρούμε ότι δεν θέλουμε $2^r \gg n$, και ότι δεν υπάρχει ζημιά αν επιλέξουμε το r όσο το δυνατόν μεγαλύτερο χωρίς να παραβιάσουμε τον παραπάνω περιορισμό.

Επιλέγοντας $r = \lg n$ συνεπάγεται $T(n, b) = \Theta(bn/\lg n)$.

- Για αριθμούς στο εύρος $0 \dots n^d - 1$, έχουμε $b = d \lg n$
 \Rightarrow η αριθμοτακτική ταξινόμηση τρέχει σε χρόνο $\Theta(d n)$