# Κατακερματισμός

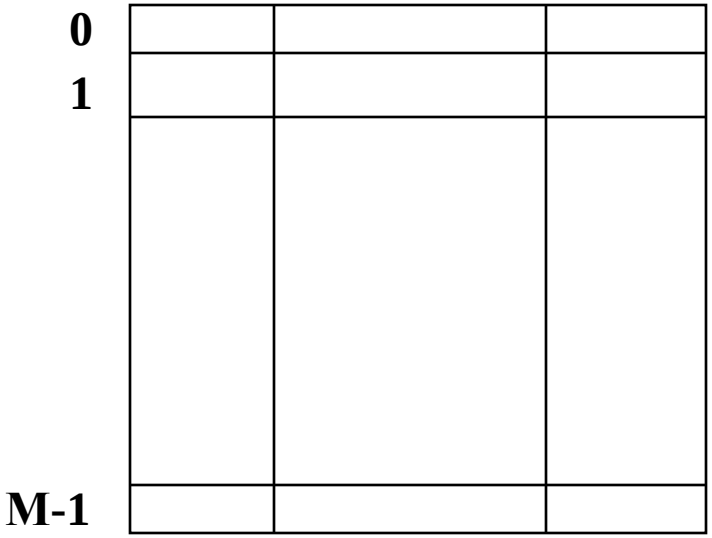
**Η ιδέα που βρίσκεται πίσω από την τεχνική του κατακερματισμού είναι να δίνεται μια συνάρτησης h, που λέγεται συνάρτηση κατακερματισμού ή παραγωγής τυχαίων τιμών (hash ή randomizing function), η οποία εφαρμοζόμενη στην τιμή του πεδίου κατακερματισμού μιας εγγραφής επιστρέφει τη διεύθυνση του μπλοκ του δίσκου στο οποίο βρίσκεται η εγγραφή αποθηκευμένη.**

**Με χρήση μιας συνάρτησης που ονομάζεται συνάρτηση απεικόνισης ή συνάρτηση κατακερματισμού (hash function) απεικονίζεται η τιμή ενός πεδίου στο χώρο των διευθύνσεων:**

$$h(K) \rightarrow A$$

**Δεν είναι εύκολο πάντα να βρεθούν καλές συναρτήσεις. Ο κυριότερος λόγος είναι ότι το πλήθος των πιθανών τιμών ενός πεδίου είναι κατά πολύ μεγαλύτερο από τον χώρο των διευθύνσεων.**

# Εσωτερικός Κατακερματισμός



**temp ← 1 ;**
**for i ← 1 to 20 do temp ← temp * code(K[i]) mod M ;**
**hash_address  temp mod M ;**

**i ← hash_address(K); a←i ;**
**if (η θέση i είναι κατειλημμένη)**
**then begin i ← (i+1) mod M ;**
**while (i ≠ a) and (η θέση i είναι κατειλημμένη)**
**do i ← (i+1) mod M ;**
**if (i = a) then όλες οι θέσεις είναι γεμάτες**
**else new_hash_address ← i**
**end ;**

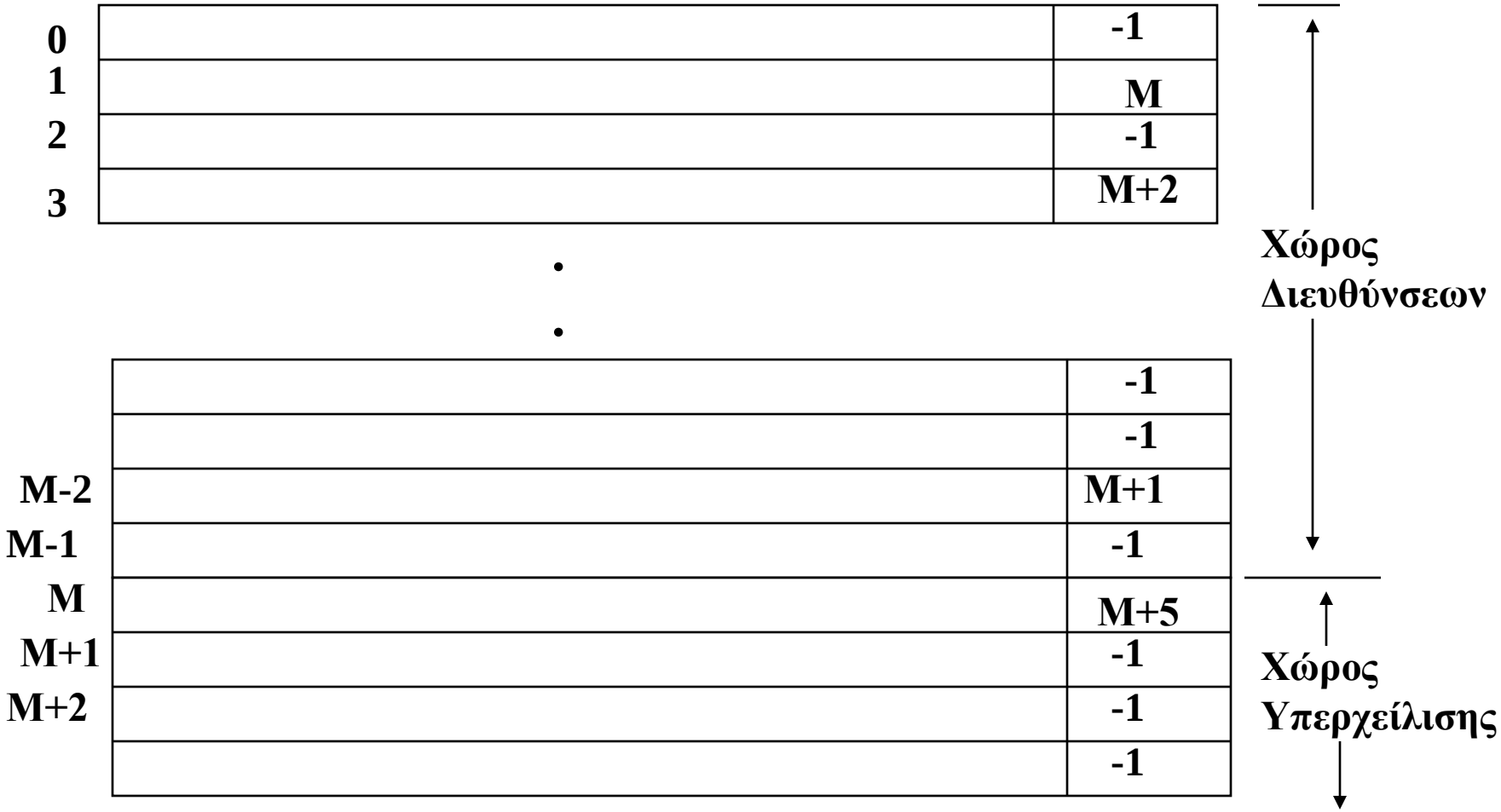
## Σύγκρουση

**Μια σύγκρουση (collision) συμβαίνει όταν η τιμή του πεδίου κατακερματισμού μιας νέας εγγραφής που εισάγεται κατακερματίζεται σε μια διεύθυνση που ήδη περιέχει μια διαφορετική εγγραφή. Στην περίπτωση αυτή πρέπει να εισάγουμε τη νέα εγγραφή σε μια άλλη θέση, αφού η διεύθυνση κατακερματισμού της είναι κατειλημμένη.**

**Ο κατακερματισμός γενικά δεν διατηρεί την διάταξη.**

## Επίλυση Συγκρούσεων

- **Ανοικτή διευθυντιοδότηση (Open Addressing)**
- **Αλυσιδωτή Σύνδεση (Chaining)**
- **Πολλαπλός Κατακερματισμός**

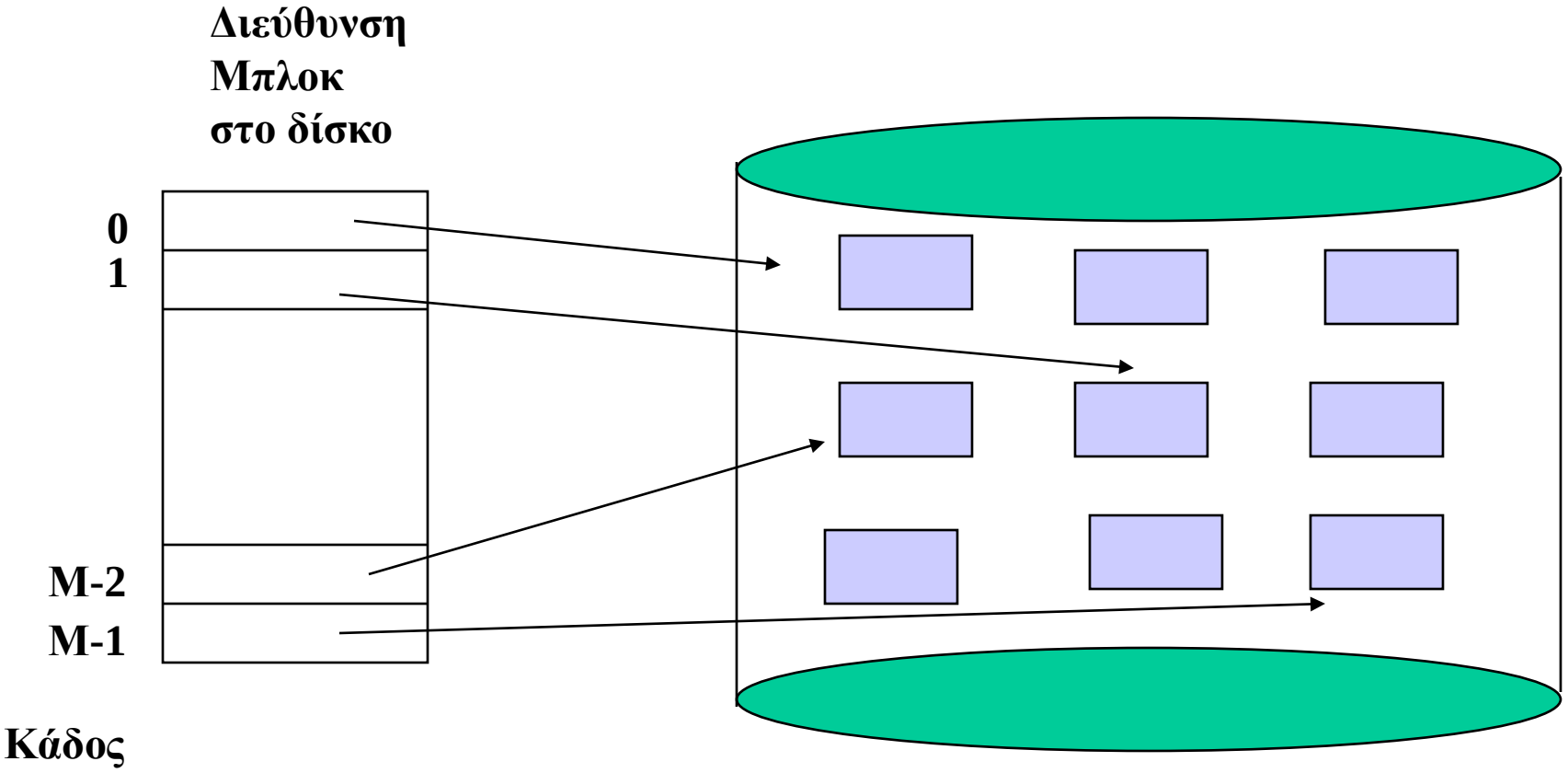

0
1
2
3
-1
M
-1
M+2
Χώρος Διευθύνσεων
M-2
M-1
M
M+1
M+2
-1
-1
M+1
-1
M+5
-1
-1
-1
Χώρος Υπερχείλισης

**Ο στόχος μιας καλής συνάρτησης κατακερματισμού είναι να κατανέμει τις εγγραφές ομοιόμορφα στο χώρο διευθύνσεων ώστε να ελαχιστοποιούνται οι συγκρούσεις χωρίς να μένουν πολλές αχρησιμοποίητες θέσεις. Η προσομοίωση αλλά και οι αναλυτικές μέθοδοι έχουν δείξει ότι συνήθως είναι καλύτερα να διατηρείται ένας πίνακας κατακερματισμού γεμάτος σε ποσοστό 70% ως 90%, έτσι ώστε το πλήθος των συγκρούσεων να παραμένει μικρό και να μην σπαταλάμε πάρα πολύ χώρος. Επομένως, αν περιμένουμε ότι θα πρέπει να αποθηκεύσουμε r εγγραφές στον πίνακα, πρέπει να επιλέξουμε M θέσεις για τον χώρο διευθύνσεων έτσι ώστε το (r/M) να βρίσκεται μεταξύ 0.7 και 0.9. Μπορεί επίσης να είναι χρήσιμο να επιλεγεί ως M ένας πρώτος αριθμός, καθώς έχει δειχθεί ότι αυτό κατανέμει καλύτερα τις διευθύνσεις κατακερματισμού στο χώρο των διευθύνσεων όταν χρησιμοποιείται ως συνάρτηση κατακερματισμού η mod.**

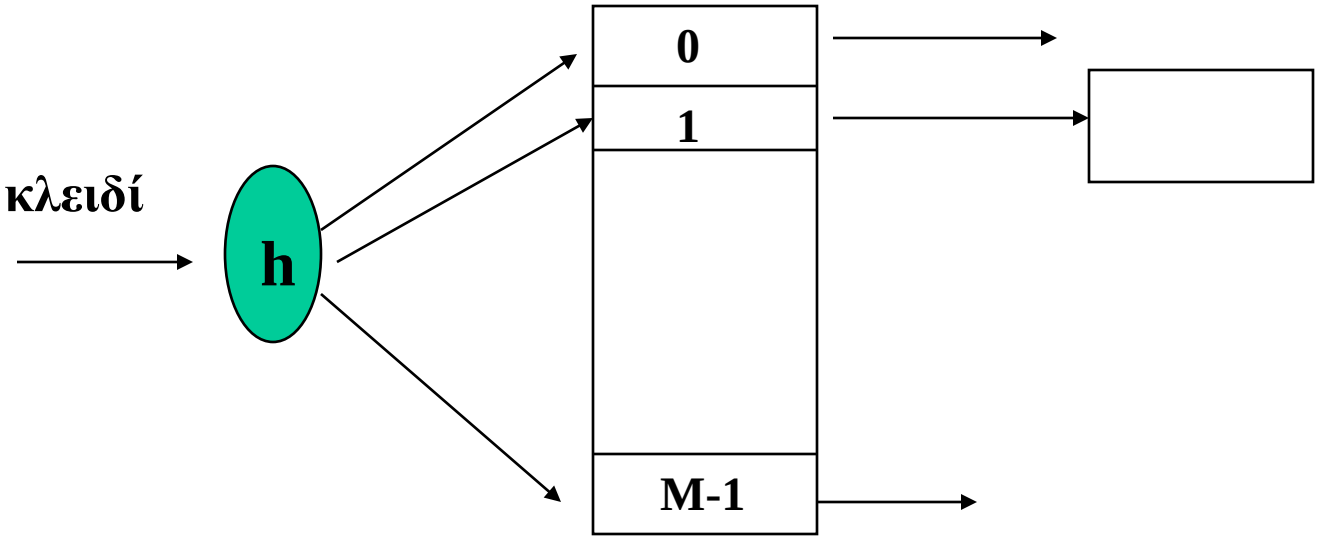
# Συναρτήσεις Κατακερματισμού

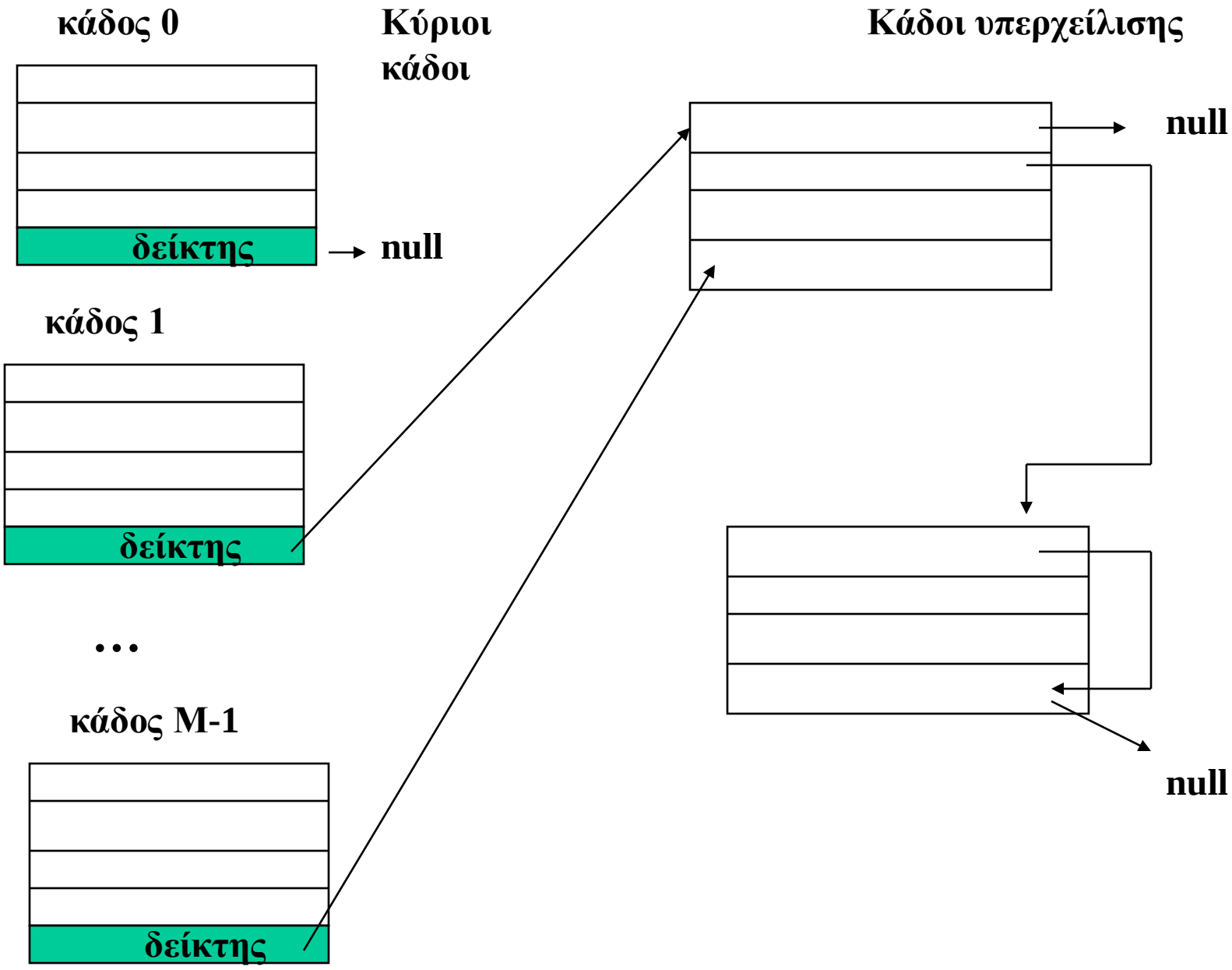
- **Μέσου Τετραγώνου**
- **Διαίρεση**
- **Αναδίπλωση**

# Εξωτερικός Κατακερματισμός



# Στατικός Εξωτερικός Κατακερματισμός




κάδος 0
Κύριοι κάδοι
Κάδοι υπερχείλισης
δείκτης
null
κάδος 1
δείκτης
...
κάδος M-1
δείκτης
null
null

**Το σχήμα αυτό ονομάζεται στατικός κατακερματισμός επειδή διατίθεται ένας σταθερός αριθμός από κάδους M. Αυτό μπορεί να είναι σοβαρό μειονέκτημα για δυναμικά αρχεία. Αν διαθέτουμε M κάδους για το χώρο διευθύνσεων και ότι m είναι το μέγιστο πλήθος εγγραφών που χωρούν σε έναν κάδο· τότε, το πολύ (m*M) εγγραφές θα χωρούν στο χώρο που διατέθηκε. Αν τελικά το πλήθος των εγγραφών είναι σημαντικά μικρότερο από (m*M), τότε έχουμε πολύ αχρησιμοποίητο χώρο. Από την άλλη πλευρά, αν το πλήθος των εγγραφών μεγαλώσει πολύ περισσότερο από (m*M), θα προκληθούν πολλές συγκρούσεις και η ανάκτηση θα επιβραδυνθεί λόγω της εμφάνισης μεγάλων λιστών από εγγραφές υπερχείλισης.**

## Τι γίνεται με

- **Αναζήτηση εγγραφής όταν δεν δίδεται η τιμή κατακερματισμού.**
- **Διαγραφές**
- **Τροποποίηση τιμής Πεδίου.**

## Δυναμικά Σχήματα Κατακερματισμού

**Τα σχήματα αυτά επιτρέπουν την δυναμική επέκταση και συρρίκνωση των αρχείων κατακερματισμού.**

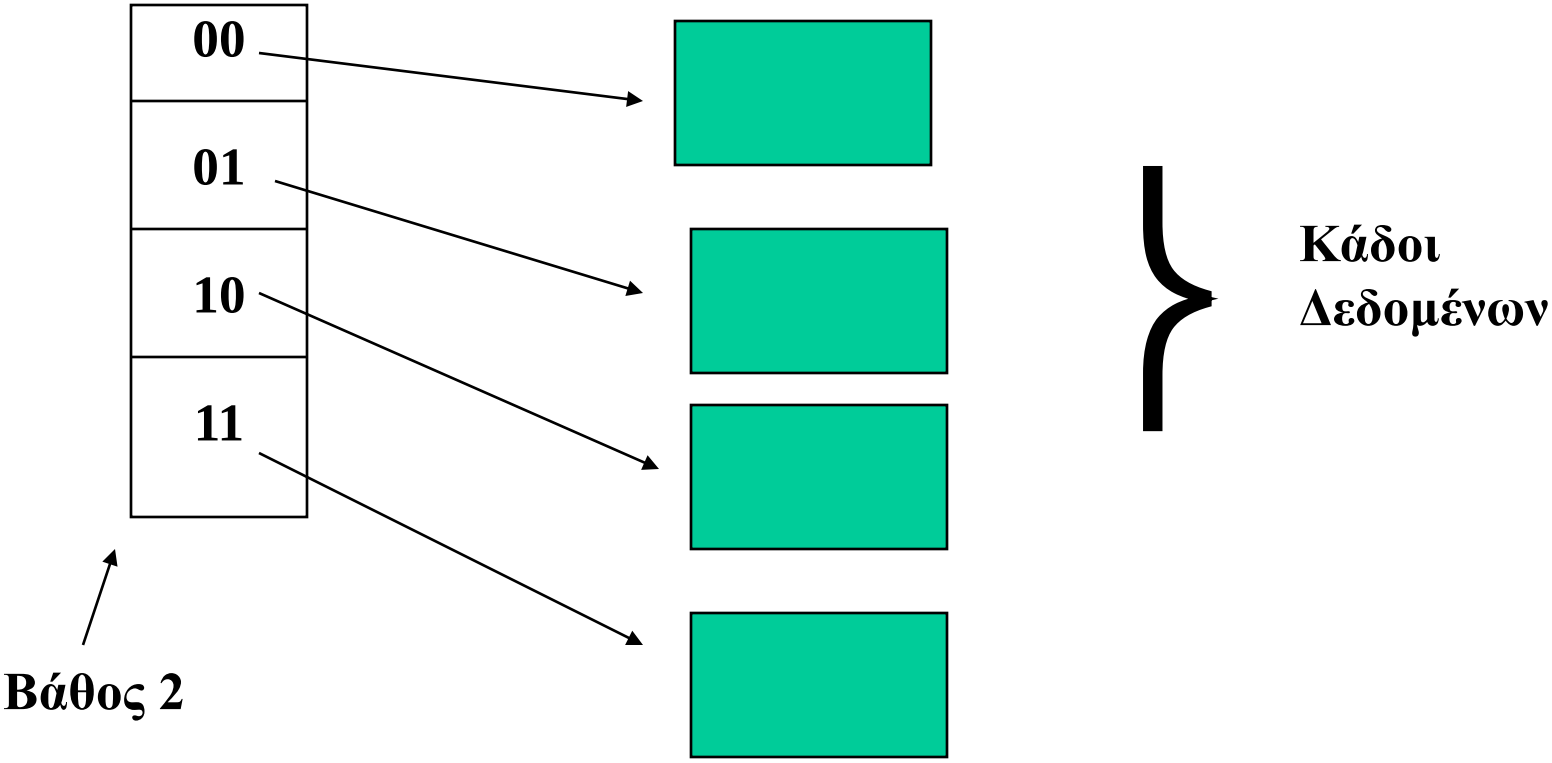
**Ο μετασχηματισμός στηρίζεται στην δυαδική αναπαράσταση την τιμής κατακερματισμού.**

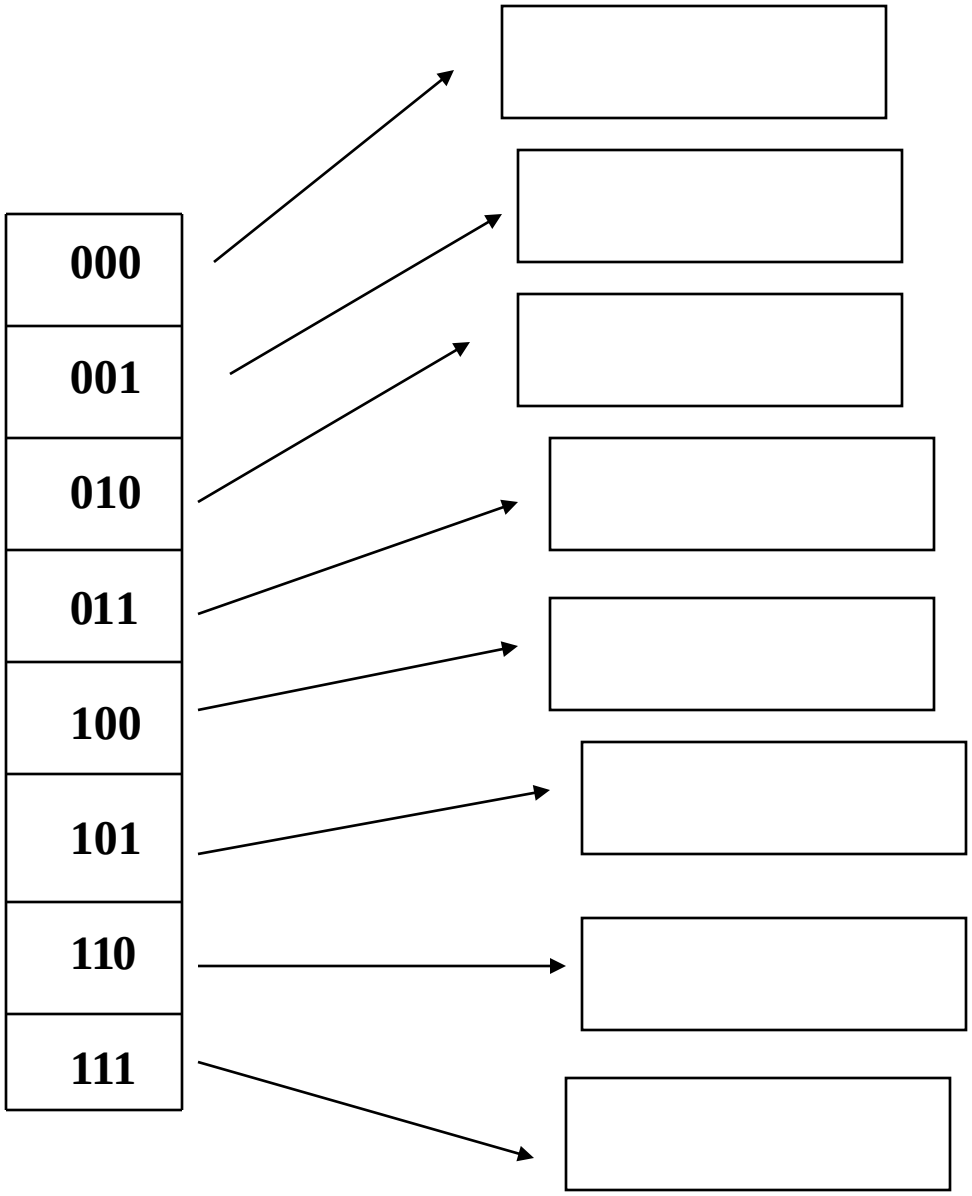
**Τα πιο γνωστά σχήματα είναι:**

**Ο επεκτατός κατακερματισμός και**

**Ο γραμμικά Επεκτατός Κατακερματισμός**

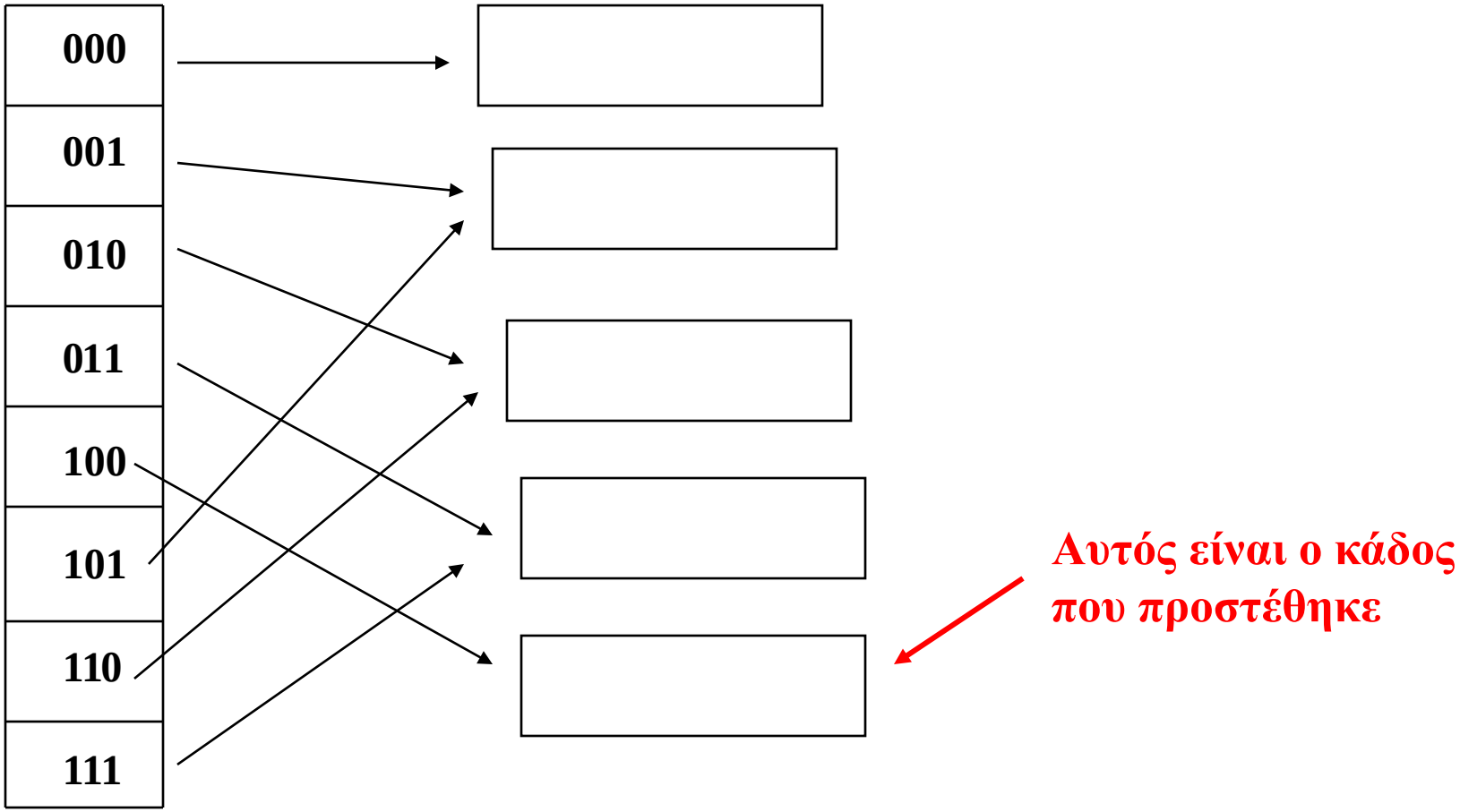
**Οι αρχές προέρχονται από τον εκθετικά επεκτατό κατακερματισμό. Μέσω της συνάρτησης κατακερματισμού το κλειδί απεικονίζεται σε μια ακέραια τιμή με δυαδική αναπαράσταση.**




000
001
010
011
100
101
110
111

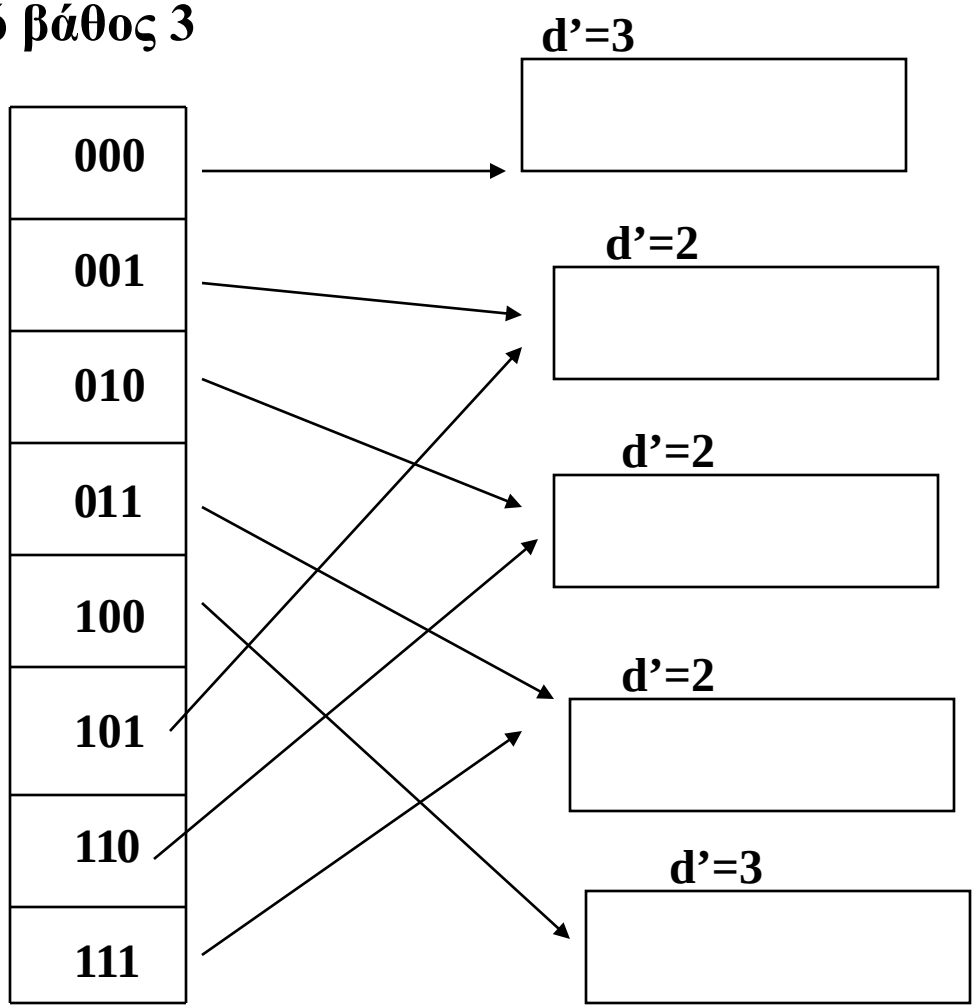
**Συνεχίζοντας με αυτόν τον τρόπο θα έχουμε άχρηστη εκθετική αύξηση του χώρου.**
**Προσπαθούμε να βρούμε τεχνικές για να περιορίσουμε την επέκταση του χώρου.**

# Η επέκταση γίνεται εκθετικά μόνο στο ευρετήριο

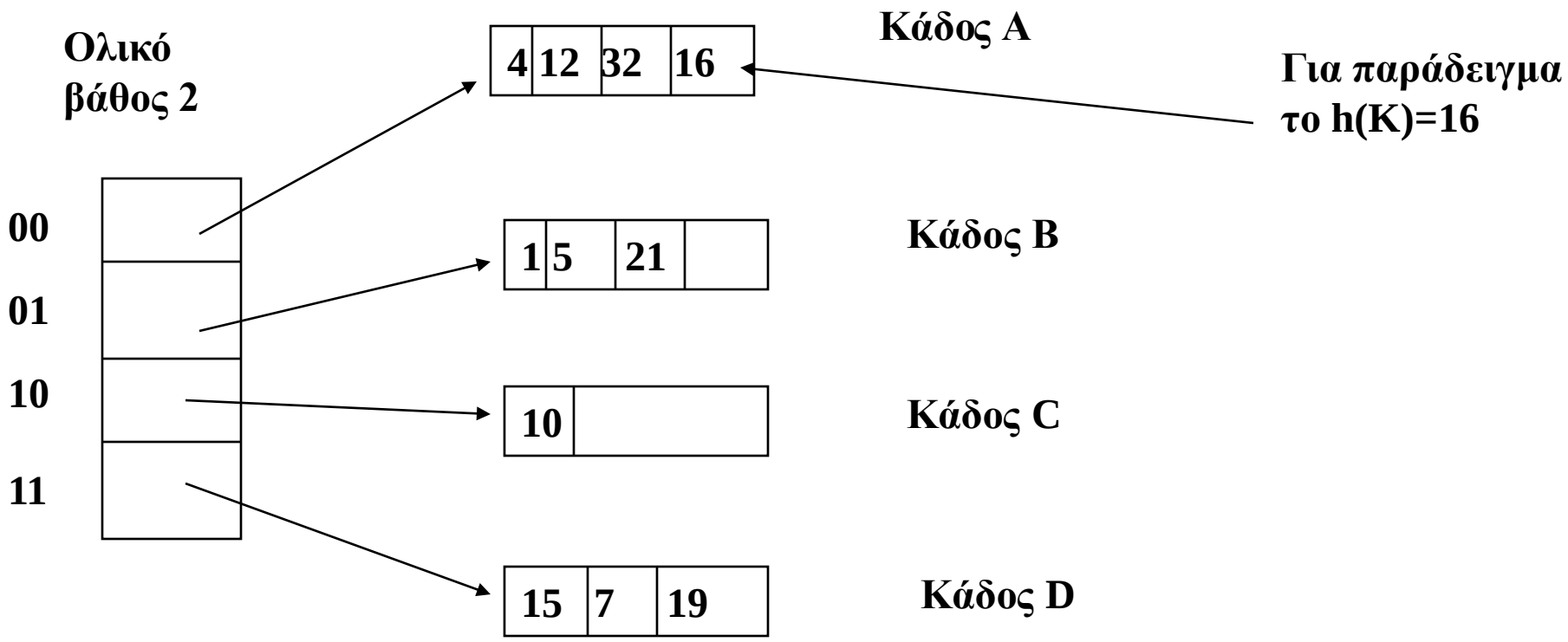



**Βάθος 3**

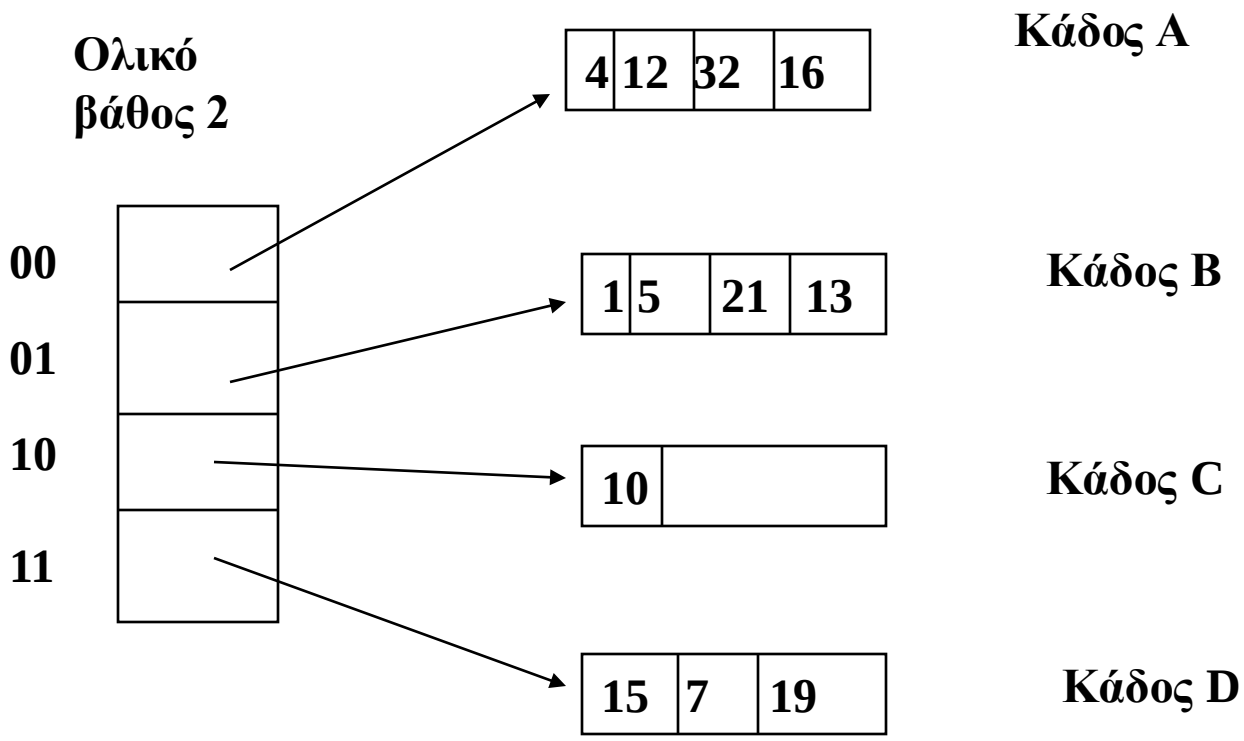
**Ολικό βάθος 3**



**Τοπικό βάθος σε όλα 2**



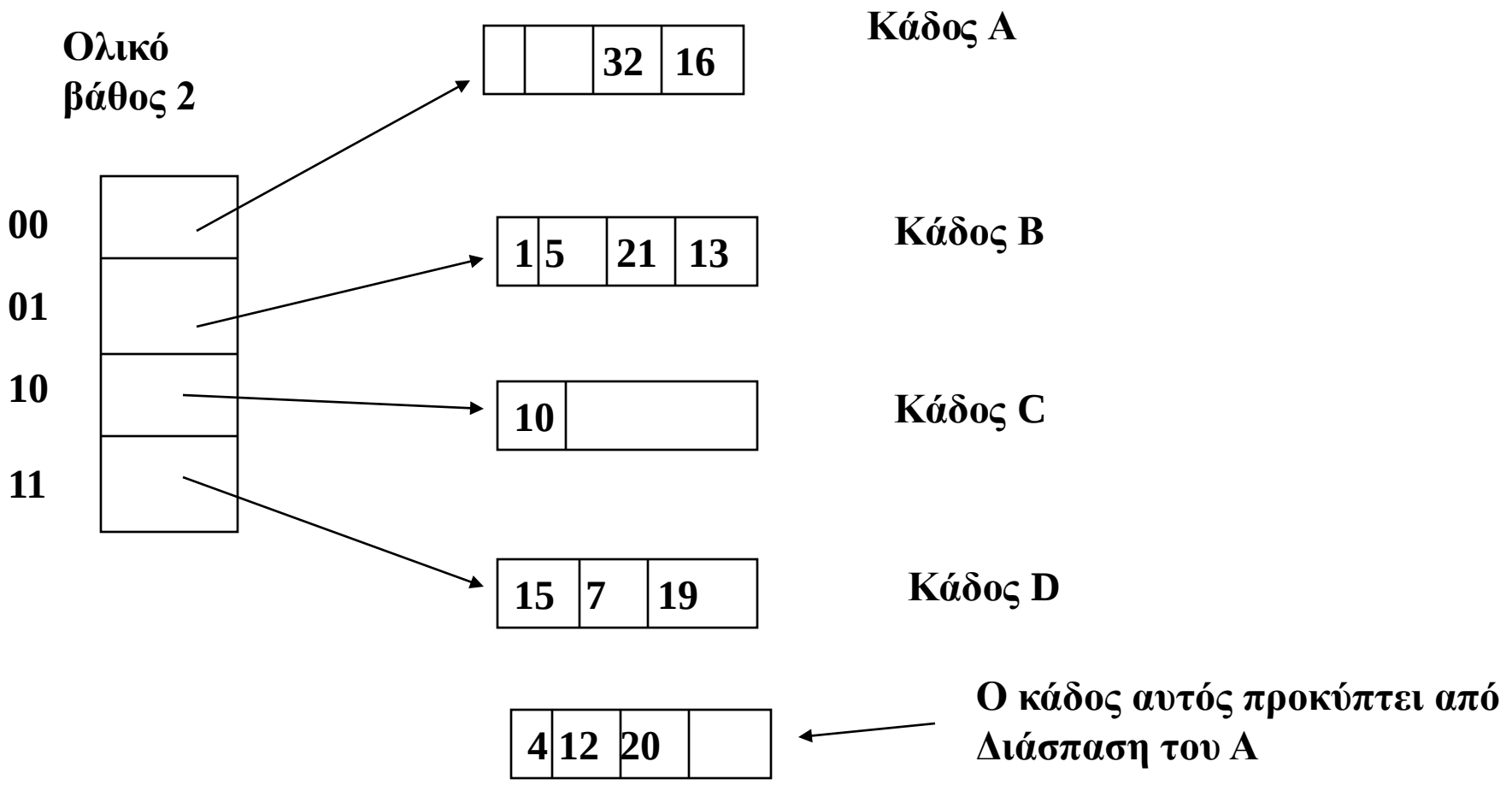
**Τοπικό βάθος σε όλα 2**

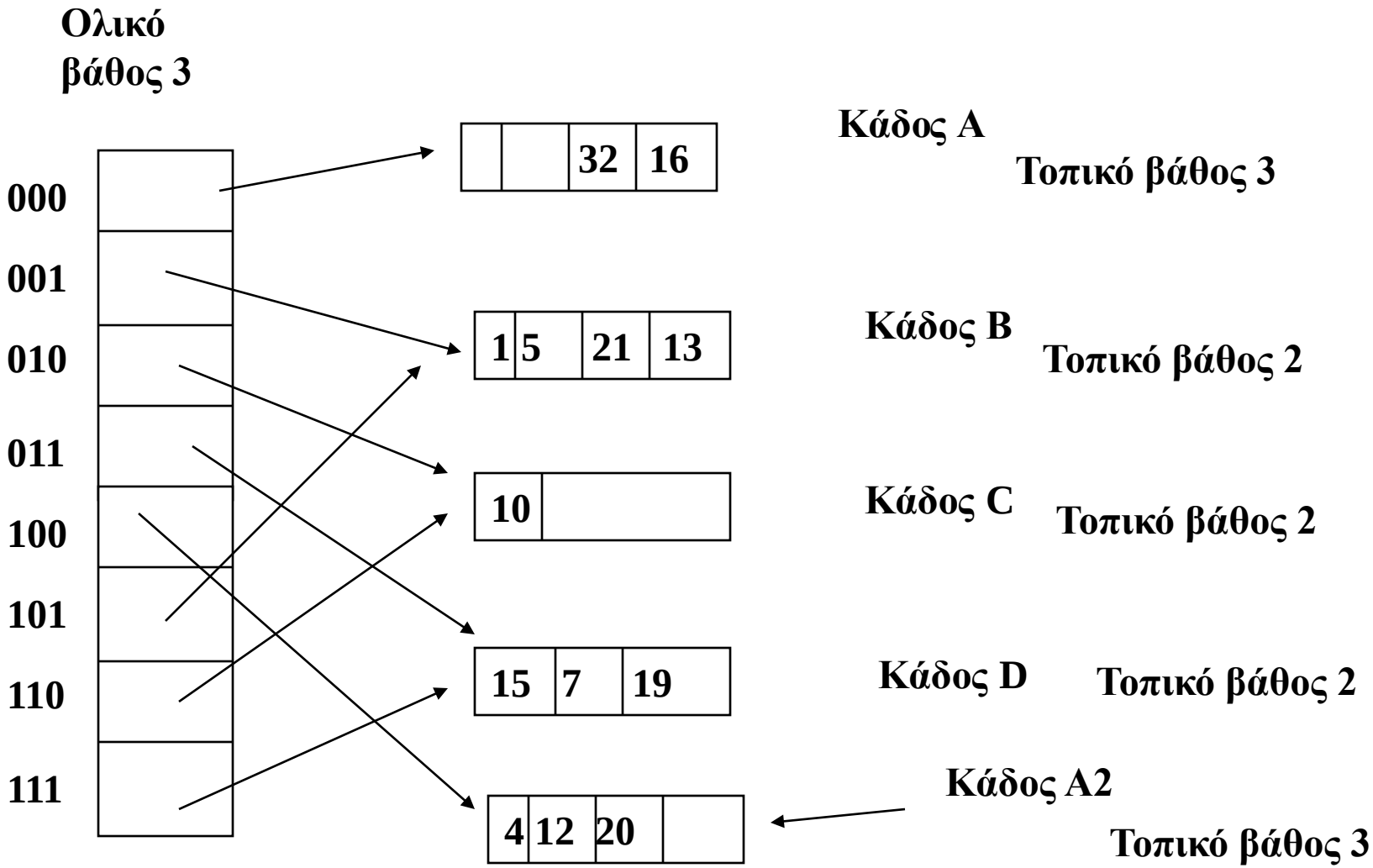



**Εισαγωγή του h(r)=13**

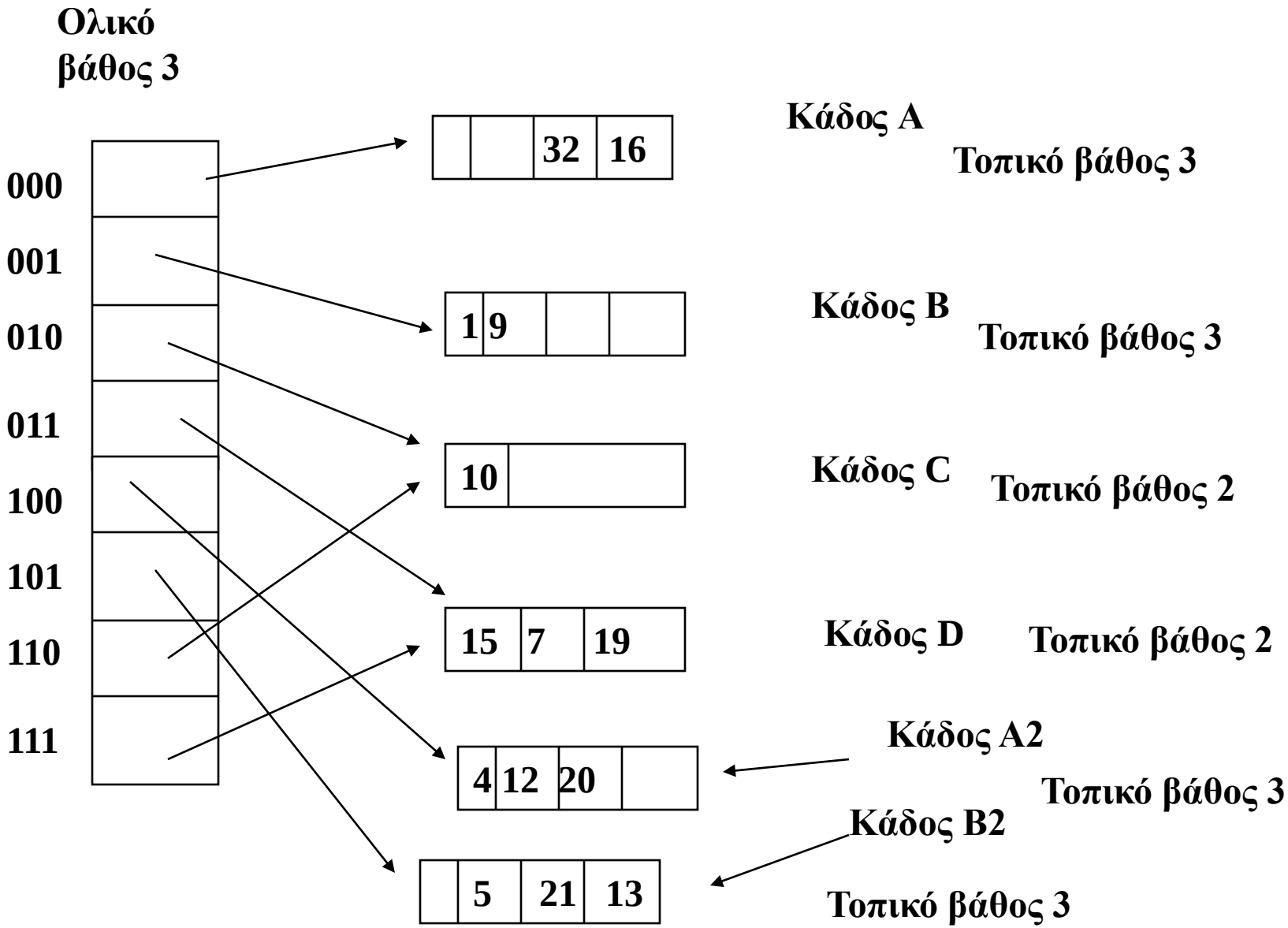
**Στη συνέχεια η εισαγωγή του 20 δημιουργεί πρόβλημα**

**Τοπικό βάθος σε όλα 2**






**Προσθήκη του 9 δημιουργεί πρόβλημα στο B**


Ολικό
βάθος 3
000
001
010
011
100
101
110
111
32
16
Κάδος A
Τοπικό βάθος 3
1
9
Κάδος B
Τοπικό βάθος 3
10
Κάδος C
Τοπικό βάθος 2
15
7
19
Κάδος D
Τοπικό βάθος 2
4
12
20
Κάδος A2
Τοπικό βάθος 3
5
21
13
Κάδος B2
Τοπικό βάθος 3

Ερώτηση
Αν το ολικό βάθος του ευρετηρίου είναι d bits και το τοπικό βάθος ενός κάδου είναι l bits πόσες καταχωρήσεις ευρετηρίου δείχνουν στον κάδο αυτό;

**Για διαγραφές εντοπίζεται ο κάδος που περιέχει το δεδομένο και ακολουθεί διαγραφή. Αν η διαγραφή οδηγεί σε κενό κάδο τότε μπορεί να συγχωνευθεί με τον κάδο από τον οποίο προήλθε από την διάσπαση. Η συγχώνευση των κάδων μειώνει το τοπικό βάθος.**
**Αυτό μπορεί να οδηγήσει και σε μείωση του ευρετηρίου στο μισό.**

**Αν το ευρετήριο χωράει στη μνήμη μια αναζήτηση ισότητας μπορεί να απαντηθεί με μια προσπέλαση στο δίσκο.**
**Αρχείο 100MB με 100 bytes καταχώρηση και μέγεθος σελίδας 4KB περιέχει 1000000 καταχωρήσεις και μόνο περίπου 25000 καταχωρήσεις ευρετηρίου. Επομένως σε πολλές περιπτώσεις μπορεί να χωράει το ευρετήριο στη μνήμη.**

## Γραμμικός Κατακερματισμός

**Ο γραμμικός κατακερματισμός είναι μια τεχνική που προσαρμόζεται με επιτυχία σε εισαγωγές και διαγραφές.**

**Η ιδέα στην οποία βασίζεται ο γραμμικός κατακερματισμός είναι να επιτρέπουμε σε ένα αρχείο κατακερματισμού να επεκτείνει ή να συρρικνώνει τους κάδους του δυναμικά χωρίς να χρειάζεται κάποιον κατάλογο.**

**Με την μέθοδο αυτή υπάρχει μια ακολουθία συναρτήσεων κατακερματισμού $h_0$, $h_1$, $h_2$,.. με την ιδιότητα το πεδίο τιμών καθεμιάς να είναι διπλάσιο από αυτό της προηγουμένης της.**

$$h_i(\text{τιμή}) = h(\text{τιμή}) \bmod (2^i N)$$

**Συνήθως το Ν είναι δύναμη του 2**

**Μπορούμε να θεωρήσουμε γύρους κατακερματισμού. Κατά τον γύρο i χρησιμοποιούνται μόνο οι συναρτήσεις $h_i$ και $h_{i+1}$. Όπως γεμίζουν οι κάδοι διαχωρίζονται ένας ένας από τον πρώτο στον τελευταίο.**

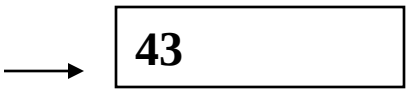
Έστω ότι η χωρητικότητα του κάδου είναι 4

Σελίδα Υπερχείλισης

Εισαγωγή 43

| h1 | h0 | | | | |
|---|---|---|---|---|---|
| 000 | 00 | 32 | 44 | 36 | _ |
| 001 | 01 | 9 | 25 | 5 | _ |
| 010 | 10 | 14 | 18 | 10 | 30 |
| 011 | 11 | 31 | 35 | 7 | 11 |

| h1 | h0 | | | | |
|---|---|---|---|---|---|
| 000 | 00 | 32 | _ | _ | _ |
| 001 | 01 | 9 | 25 | 5 | _ |
| 010 | 10 | 14 | 18 | 10 | 30 |
| 011 | 11 | 31 | 35 | 7 | 11 |
| 100 | 00 | 44 | 36 | | |



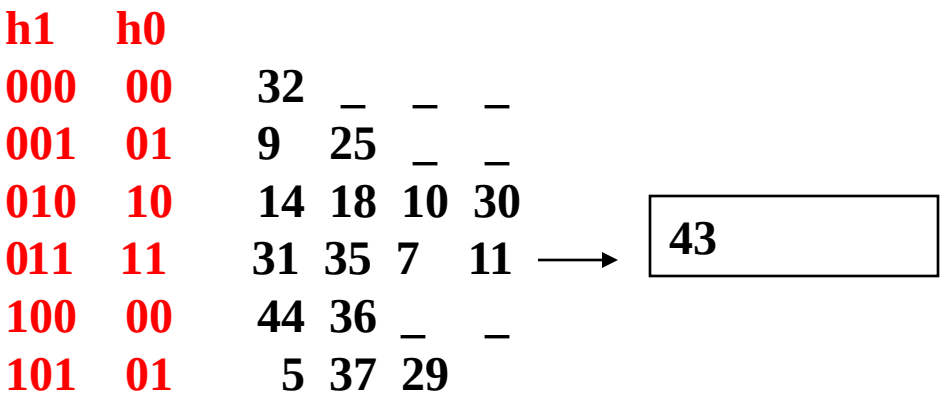

Τα κόκκινα πληροφοριακά μόνο

## Διάσπαση συμβαίνει όταν γίνει επέκταση στην υπερχείλιση

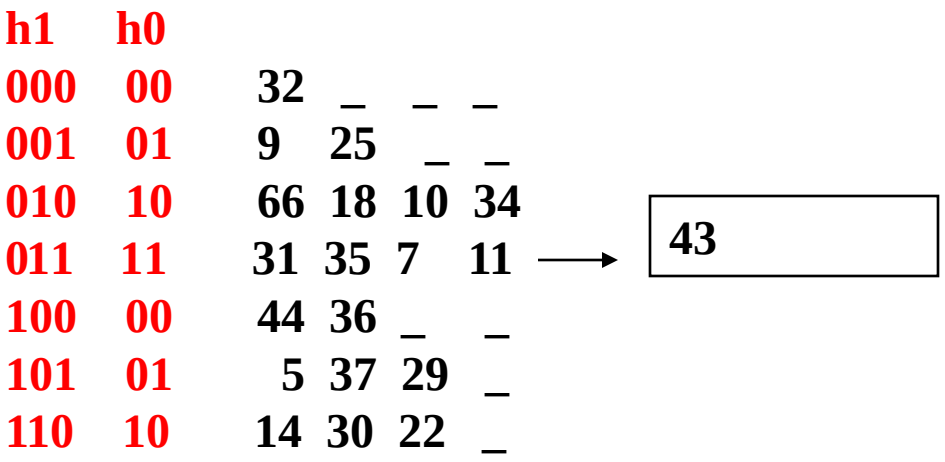
**Εισαγωγή 37**

| h1 | h0 | | |
|---|---|---|---|
| 000 | 00 | 32 _ _ _ | |
| 001 | 01 | 9 25 5 37 | |
| 010 | 10 | 14 18 10 30 | |
| 011 | 11 | 31 35 7 11 | → 43 |
| 100 | 00 | 44 36 | |

## Εισαγωγή 29

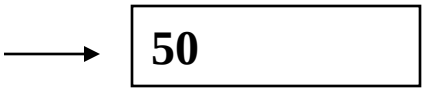

h1 h0
000 00 32 _ _ _
001 01 9 25 _ _
010 10 14 18 10 30
011 11 31 35 7 11
100 00 44 36 _ _
101 01 5 37 29
43

**Εισαγωγή των 22 66 και 34**



## Εισαγωγή 50

| h1 | h0 | | | | |
|---|---|---|---|---|---|
| 000 | 00 | 32 | _ | _ | _ |
| 001 | 01 | 9 | 25 | _ | _ |
| 010 | 10 | 66 | 18 | 10 | 34 |
| 011 | 11 | 43 | 35 | 11 | _ |
| 100 | 00 | 44 | 36 | _ | _ |
| 101 | 01 | 5 | 37 | 29 | _ |
| 110 | 10 | 14 | 30 | 22 | _ |
| 111 | 11 | 31 | 7 | _ | _ |



**Αλγόριθμος Η διαδικασία αναζήτησης για γραμμικό κατακερματισμό.**

**if (n = 0)**

**then m ← $h_j$(K) (* το m είναι η τιμή κατακερματισμού της εγγραφής με κλειδί κατακερματισμού K *)**

**else begin**

**m ← $h_j$(K) ;**

**if (m < n) then m ← $h_{j+1}$(K)**

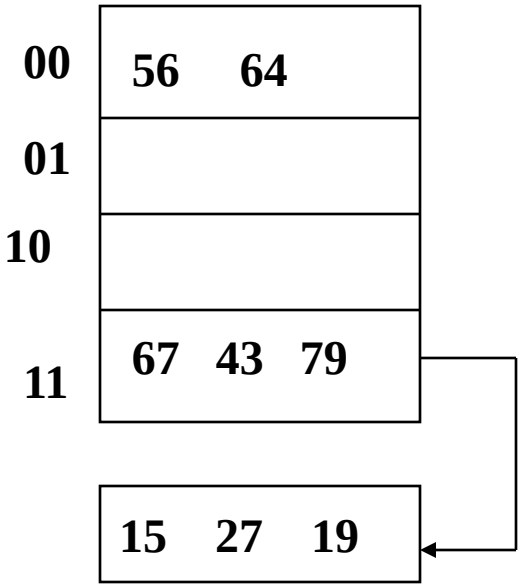
**end;**

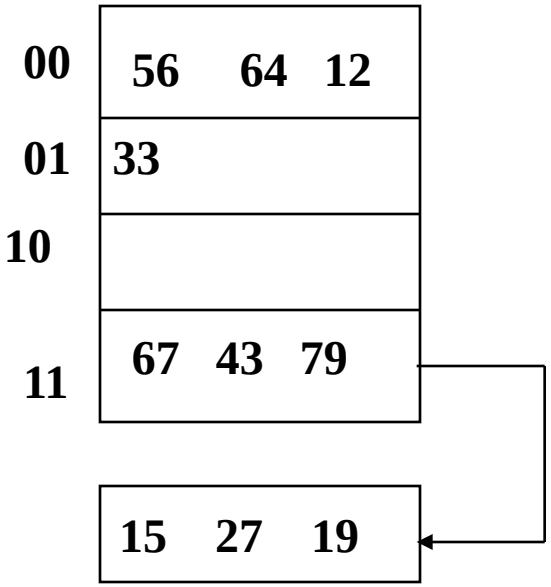
**αναζήτησε τον κάδο που η τιμή κατακερματισμού του είναι m (και την υπερχείλισή του, αν έχει) ;**

**Μια άλλη πολιτική διάσπασης θα ήταν ο παράγοντας φόρτωσης. Όταν ξεπεράσει μια τιμή να γίνεται διάσπαση.**

**Παράγοντας φόρτωσης l=r/(bfr*N)**
**r: το πλήθος των εγγραφών του αρχείου**
**bfr: το μέγιστο πλήθος εγγραφών στον κάδο**
**N: το τρέχον πλήθος των κάδων**

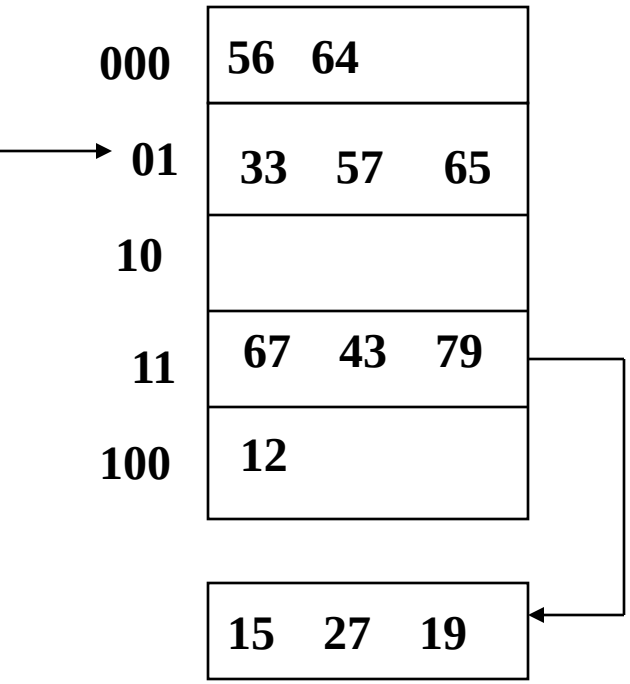
**Αν υποθέσουμε ότι κάθε κάδος έχει 3 θέσεις και παράγοντα φόρτωσης του αρχείου 67%**



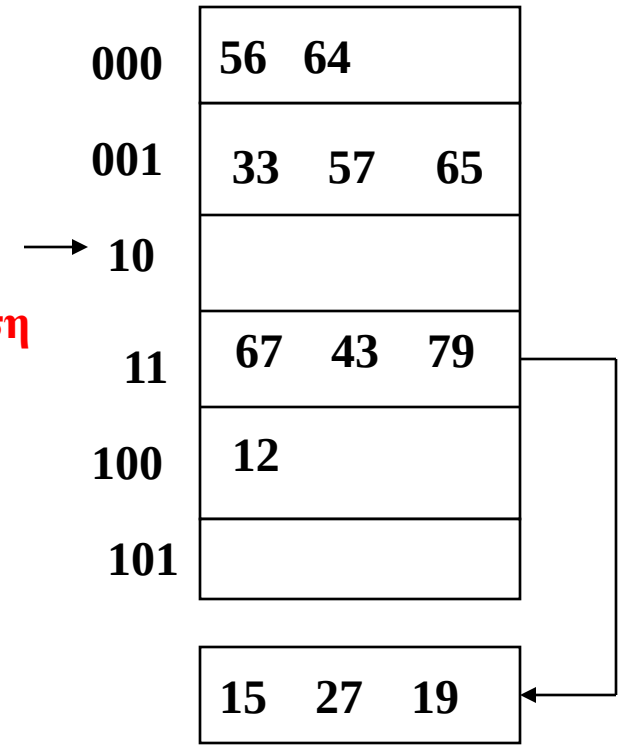




56=0111000
67=1000011
43=0101011
79=1001111
15=0001111
27=0011011
19=0010011
64=1000000
12=0001100
33=0100001
57=0111101
65=1000001

**Στην επόμενη εισαγωγή θα πρέπει να γίνει διάσπαση**




**Για την επόμενη εισαγωγή**
**Θα χρειασθεί διάσπαση**



**Γενικά μπορούμε να εφαρμόσουμε και την αντίστροφη διαδικασία συρρίκνωσης του αρχείου όταν διαπιστώσουμε μεγάλο αχρησιμοποίητο χώρο. Για παράδειγμα αν ο παράγοντας φόρτωσης πέσει κάτω από μια τιμή.**