Κεφάλαιο 10

ΔΥΝΑΜΙΚΑ ΤΥΧΑΙΑ ΑΡΧΕΙΑ

Κεφάλαιο 10

ΔΥΝΑΜΙΚΑ ΤΥΧΑΙΑ ΑΡΧΕΙΑ

10.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο εξετάσθηκαν τα τυχαία στατικά αρχεία. Ο αναλυτής/προγραμματιστής κατά τη φάση σχεδιασμού και φόρτωσης των αρχείων αυτών πρέπει να αποφασίσει πόσος χώρος θα καταληφθεί συνολικά στην κύρια περιοχή και στην περιοχή υπερχείλισης. Σε πολλές εφαρμογές είναι δυνατόν ο αριθμός των εγγραφών να ποικίλει κατά πολύ. Απεναντίας, ο χώρος του αρχείου μένει αμετάβλητος ανεξάρτητα αν το αρχείο έχει πολύ χαμηλή ή πολύ υψηλή πληρότητα, για παράδειγμα 5% ή 95%. Στην πρώτη περίπτωση υπάρχει πρόβλημα αχρησιμοποίητου χώρου, ενώ στη δεύτερη πρόβλημα φτωχής επίδοσης κατά την αναζήτηση. Τα στατικά αρχεία διακρίνονται για αυτές τις αδυναμίες μέχρι του σημείου που θα γίνει αναδιοργάνωση. Κατά την αναδιοργάνωση όλο το αρχείο σχεδιασμένο από την αρχή θα ξαναγραφεί σε μικρότερο ή σε μεγαλύτερο χώρο και μάλιστα σε άλλο μέρος του δίσκου. Ωστόσο, η διαδικασία αυτή είναι ιδιαίτερα χρονοβόρα. Σε μερικές εφαρμογές αυτό μπορεί να είναι απαράδεκτο επειδή το αρχείο δεν είναι διαθέσιμο στο χρήστη κατά το συγκεκριμένο χρονικό διάστημα.

Στο παρελθόν έχουν προταθεί πολλές οργανώσεις τυχαίων αρχείων, που δεν παραμένουν παθητικές στις εισαγωγές και διαγραφές εγγραφών αλλά προσαρμόζονται, ώστε και να μη γίνεται σπατάλη χώρου αλλά και η επίδοση να μην εκφυλίζεται. Στη βιβλιογραφία αναφέρονται περί τις δέκα παραλλαγές τέτοιων τυχαίων αρχείων που λέγονται **δυναμικές** (dynamic). Στη συνέχεια θα εξετασθούν από αλγοριθμική ή/και αναλυτική άποψη οι εξής

οργανώσεις: ο **δυναμικός κατακερματισμός** (dynamic hashing), ο **επεκτατός κατακερματισμός** (extendible hashing), ο **εκθετικός κατακερματισμός με περιορισμένο κατάλογο** (bounded index exponential hashing), και τέλος ο **γραμμικός κατακερματισμός** (linear hashing). Κάθε μία από τις προηγούμενες τεχνικές με τα προτερήματα και τα μειονεκτήματα σε σχέση με τις υπόλοιπες απαντά με σύνθετους αλγορίθμους στα προβλήματα:

- πως και πότε διασπάται ένας κάδος,
- πως διανέμονται οι εγγραφές από τον παλιό κάδο στους νέους,
- πως και πότε συγχωνεύονται δύο κάδοι, και τέλος
- πως οι εγγραφές από τους δύο παλιούς κάδους αποδίδονται στο νέο κάδο.

10.2 Δυναμικός κατακερματισμός

Ο δυναμικός κατακερματισμός ήταν η πρώτη χρονικά δομή δυναμικών τυχαίων αρχείων που εμφανίσθηκε στη βιβλιογραφία. Ο ερευνητής που την παρουσίασε, ο Larson (1978), θεωρείται από τους θεμελιωτές της περιοχής αυτής. Ο δυναμικός κατακερματισμός αποτελείται από δύο φυσικά ανεξάρτητες δομές: έναν κατάλογο και ένα κυρίως αρχείο. Ο κατάλογος είναι ένα δάσος δυαδικών δένδρων που υλοποιούνται ως συνδεδεμένες δομές και είναι αποθηκευμένα στην κύρια μνήμη. Στο τελευταίο επίπεδο των δυαδικών δένδρων περιέχονται δείκτες προς τις σελίδες του κυρίως αρχείου που βέβαια είναι αποθηκευμένο στη δευτερεύουσα μνήμη. Ο αριθμός των δυαδικών δένδρων που αποτελούν το δάσος ισούται με τον αριθμό των κάδων, bk, του αρχείου όπως έχει σχεδιασθεί αρχικά. Μία συνάρτηση κατακερματισμού $h_1(key) = key \bmod bk$ από την τιμή του κλειδιού δίνει το δυαδικό δένδρο, όπου είναι αποθηκευμένη η διεύθυνση του κάδου όπου η εγγραφή είναι πραγματικά αποθηκευμένη. Αυτή είναι η γενική φιλοσοφία λειτουργίας της δομής που στηρίζεται στη στρατηγική `διαίρει και βασίλευε´. Στη συνέχεια αναλύονται δύο ειδικότερα θέματα. Πρώτον, πως μεγεθύνεται ένα αρχείο και, δεύτερον, ποιά είναι η δομή και η επεξεργασία των δυαδικών δένδρων.

Κάθε κάδος από τους bk έχει χωρητικότητα $Bkfr$ εγγραφές, οπότε αν κάποιος από αυτούς δεχθεί την $(Bkfr+1)$-οστή εγγραφή τότε πρέπει να διασπασθεί. Αυτό σημαίνει ότι ένας νέος κάδος παραχωρείται από το σύστημα

στο αρχείο και οι $(Bkfr+1)$ εγγραφές πρέπει να αναδιανεμηθούν μεταξύ των δύο κάδων. Αυτό επιτυγχάνεται με τη βοήθεια μίας δεύτερης συνάρτησης κατακερματισμού $h_2(key)$ που παίρνοντας ως σπόρο το κλειδί (ή κάποιο μετασχηματισμό του κλειδιού) μπορεί να παράγει μία ψευδοτυχαία δυαδική συμβολοσειρά οποιουδήποτε μήκους, όπου τα 0 και 1 εμφανίζονται ισοπίθανα. Βέβαια αρχικά απαιτούνται μόνο μερικά bits, όμως με τις διαδοχικές διασπάσεις των κάδων χρειάζονται όλο και περισσότερα. Αυτός είναι ο λόγος που αυτή η δεύτερη συνάρτηση κατακερματισμού δεν μετατρέπει απλά το κλειδί στο δυαδικό αντίστοιχο. Έτσι, σύμφωνα με μία σύμβαση, αν το πρώτο bit είναι 0 (αντίστοιχα, 1), τότε η αντίστοιχη εγγραφή κατευθύνεται στον υπάρχοντα (αντίστοιχα, στο νέο) κάδο. Στη γενική περίπτωση, χρησιμοποιούνται τόσα bits της δυαδικής συμβολοσειράς όσα είναι απαραίτητα ώστε οι $Bkfr+1$ εγγραφές να διαμοιρασθούν σε δύο κάδους.

Αρχικά καθένα από τα bk δυαδικά δένδρα του δάσους αποτελείται μόνο από μία ρίζα, οπότε θεωρείται ότι ο κατάλογος έχει μόνο ένα επίπεδο. Όταν γίνει κάποια διάσπαση κάδου, τότε πρέπει το αντίστοιχο δένδρο να επεκταθεί κατά ένα επίπεδο. Προφανώς, δεν συμβαίνει το ίδιο σε όλα τα δένδρα, λέγεται όμως ότι ο κατάλογος επεκτείνεται κατά ένα επίπεδο. Έτσι από τη ρίζα του αντίστοιχου δένδρου δημιουργούνται δύο φύλλα με τους αντίστοιχους δείκτες προς τους δύο κάδους. Κατά σύμβαση, ο αριστερός κόμβος δείχνει στον υπάρχοντα κάδο, ενώ ο δεξιός κόμβος δείχνει στο νέο κάδο. Αυτή η διαδικασία επέκτασης των δυαδικών δένδρων γίνεται ανάλογα με τις εισαγωγές των νέων εγγραφών. Αν η συνάρτηση h_1 διασπείρει τυχαία τις εγγραφές στα δυαδικά δένδρα, τότε τα δένδρα έχουν περίπου το ίδιο μέγεθος, ενώ εκτιμάται ότι τα δένδρα είναι ισοζυγισμένα επειδή η ψευδοτυχαία συνάρτηση h_2 παράγει ισοπίθανα τα 0 και 1.

Είναι ευνόητο ότι οι τύποι των εσωτερικών και των εξωτερικών κόμβων είναι διαφορετικοί. Πιο συγκεκριμένα:

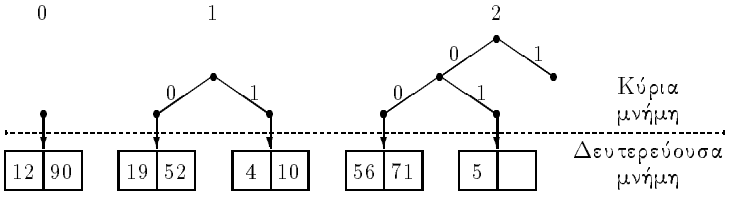
- κάθε εσωτερικός κόμβος αποτελείται από τα εξής τέσσερα πεδία: σημαία, δενδρικός δείκτης προς τον πατέρα κόμβο, δενδρικός δείκτης προς τον αριστερό απόγονο και δενδρικός δείκτης προς τον δεξιό απόγονο,
- κάθε εξωτερικός κόμβος αποτελείται από τα εξής τέσσερα πεδία: σημαία, δενδρικός δείκτης προς τον πατέρα κόμβο, δείκτης προς τον κάδο του αρχείου και μετρητής εγγραφών που είναι αποθηκευμένες στον αντίστοιχο κάδο.

Η σημαία των εσωτερικών και των εξωτερικών κόμβων παίρνει την τιμή 0 και 1, αντίστοιχα.

$R(0)$	$R(1)$	$R(2)$	$R(3)$	$R(4)$	$R(5)$	$R(6)$	$R(7)$	$R(8)$	$R(9)$
1011	0000	0100	0110	1111	0101	0001	1110	1001	0011

Πίνακας 10.1: Γεννήτρια ψευδοτυχαίων δυαδικών αριθμών.

Η προηγούμενη διαδικασία φαίνεται στο παράδειγμα που ακολουθεί. Έστω ότι σε κενό αρχείο δυναμικού κατακερματισμού με κάδους μεγέθους δύο εγγραφών πρόκειται να εισαχθούν διαδοχικά εγγραφές με τιμές κλειδιών 4, 5, 10, 12, 19, 52, 56, 71 και 90. Αρχικά το αρχείο σχεδιάζεται ώστε να έχει τρεις κάδους, άρα πρέπει ως πρώτη συνάρτηση κατακερματισμού να χρησιμοποιηθεί η συνάρτηση $h_1 = key \bmod 3$ που χωρίζει το δάσος του καταλόγου σε τρία διακριτά δυαδικά δένδρα. Πρέπει επίσης να χρησιμοποιηθεί και μία δεύτερη συνάρτηση για το κτίσιμο των δυαδικών δένδρων. Έστω, λοιπόν, ότι επιλέγεται η συνάρτηση $key \bmod 10$, ώστε από το κλειδί να προκύψει υπόλοιπο από 0 ως 9. Κάθε τιμή του υπολοίπου μπορεί να θεωρηθεί ως σπόρος σε μία ψευδοτυχαία γεννήτρια, $R()$, που να παράγει δυαδικά ψηφία 0 και 1 με την ίδια πιθανότητα (δηλαδή 50%). Το αποτέλεσμα μίας τέτοιας γεννήτριας για κάθε σπόρο φαίνεται στον Πίνακα 10.1. Στο Σχήμα 10.1 φαίνεται το τελικό αποτέλεσμα της εισαγωγής των εγγραφών με τη δημιουργία του καταλόγου των τριών δένδρων και του αρχείου των πέντε κάδων.



Σχήμα 10.1: Αρχείο δυναμικού κατακερματισμού.

Με το παράδειγμα αυτό φαίνεται ότι η επιτυχής αναζήτηση κοστίζει μία μόνο προσπέλαση στο δίσκο. Επίσης κατά την αναζήτηση μη υπάρχοντος κλειδιού μπορεί να βρεθεί φύλλο του δυαδικού δένδρου με μετρητή εγγραφών

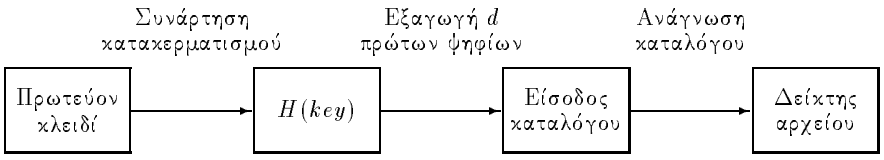
ίσο με μηδέν. Άρα στην περίπτωση αυτή το αντίστοιχο χρονικό κόστος ανεπιτυχούς αναζήτησης θα είναι μηδέν από την άποψη των προσπελάσεων στο δίσκο. Για παράδειγμα, έστω ότι γίνεται αναζήτηση της εγγραφής με τιμή κλειδιού 68. Για την αναζήτηση αυτή θα εξετασθεί το τελευταίο δυαδικό δένδρο (επειδή: 68 mod $3 = 2$), και ειδικότερα το δεξιό παιδί της ρίζας (επειδή: 60 mod $10 = 8$ και $R(8) = 1001$, το οποίο αρχίζει από 1). Παρατηρούμε, λοιπόν, ότι από το συγκεκριμένο φύλλο δεν υπάρχει δείκτης προς το δίσκο, άρα το κόστος αυτής της ανεπιτυχούς αναζήτησης είναι 0.

Ωστόσο, γίνεται αντιληπτό ότι λόγω των εισαγωγών ο κατάλογος αυξάνει βαθμιαία, οπότε μπορεί το μέγεθός του να ξεπεράσει το μέγεθος της διαθέσιμης κύριας μνήμης. Έτσι σε κάποια χρονική στιγμή η επιτυχής αναζήτηση θα κοστίζει δύο προσπελάσεις στο δίσκο γιατί τμήμα του καταλόγου θα πρέπει να αποθηκευθεί στη δευτερεύουσα μνήμη. Σε σχέση με τον παράγοντα χρησιμοποίησης χώρου έχει αποδειχθεί αναλυτικά ότι η μέση τιμή του είναι 69%. Η τιμή αυτή ισούται με την αντίστοιχη για τις δομές της οικογένειας των B-δένδρων και θα συναντηθεί και πάλι στη συνέχεια του κεφαλαίου αυτού.

Από τον Scholl (1981) έχουν προταθεί δύο παραλλαγές του δυναμικού κατακερματισμού για τη βελτίωση της επίδοσης της αναζήτησης. Σύμφωνα με μία παραλλαγή από αυτές, δεν γίνεται διάσπαση του κάδου όταν σ΄ αυτόν κατευθυνθεί η $(Bkfr+1)$-οστή εγγραφή αλλά δημιουργείται αλυσίδα υπερχείλισης με ένα δεύτερο κάδο. Το ζεύγος αυτό των κάδων θα διασπασθεί και ο κατάλογος θα ενημερωθεί αντίστοιχα όταν θα έχουν εισαχθεί $\beta \times Bkfr$ εγγραφές, όπου $1 \leq \beta \leq 2$. Η τεχνική αυτή λέγεται **αναβολή διάσπασης** (deferred splitting). Έτσι βέβαια για μερικές περιπτώσεις θα χρειάζεται μία προσπέλαση στο δίσκο, ενώ για άλλες περιπτώσεις θα χρειάζονται δύο προσπελάσεις. Το πλεονέκτημα όμως είναι ότι ο κατάλογος θα είναι μικρότερος κατά β φορές και θα χωρά στην κύρια μνήμη. Αν το αρχείο μεγαλώσει υπερβολικά τότε μπορεί δίνοντας στο β μεγαλύτερες τιμές να επιτραπούν αλυσίδες υπερχείλισης μεγαλύτερου μήκους, ώστε ο κατάλογος να χωρά οπωσδήποτε στην κύρια μνήμη. Εύκολα συμπεραίνεται ότι η μέση τιμή προσπελάσεων στο δίσκο στη χειρότερη περίπτωση θα είναι $\lceil \beta \rceil$. Ο ενδιαφερόμενος αναγνώστης μπορεί από την αναφορά να βρει περισσότερες λεπτομέρειες για τις παραλλαγές του δυναμικού κατακερματισμού.

10.3 Επεκτατός κατακερματισμός

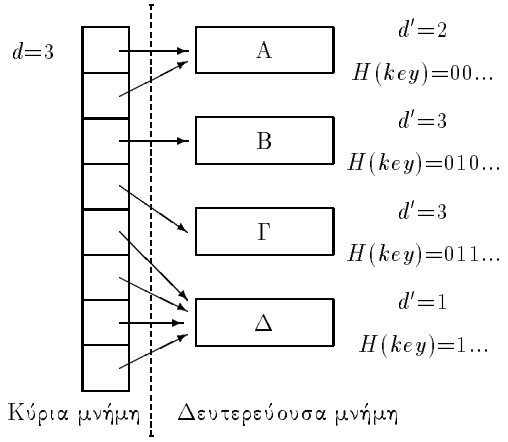
Η δομή αυτή προτάθηκε από τους Fagin et al. (1979) και ήταν η χρονικά δεύτερη υλοποίηση δυναμικού τυχαίου αρχείου. Η αλληλουχία των μετασχηματισμών του κλειδιού που εκτελούνται στον επεκτατό κατακερματισμό φαίνεται στο Σχήμα 10.2. Ο πρώτος μετασχηματισμός είναι μία συνάρτηση κατακερματισμού που απεικονίζει κατά τυχαίο τρόπο ένα διάστημα κλειδιών σε ένα σταθερό διάστημα διευθύνσεων. Ο μόνος περιορισμός είναι ότι το διάστημα διευθύνσεων πρέπει να είναι δύναμη του δύο. Επειδή όμως προτιμάται το διάστημα διευθύνσεων να είναι πρώτος αριθμός, γι΄ αυτό τελικά επιλέγεται ο μεγαλύτερος ακέραιος που είναι αμέσως μικρότερος από τη δυαδική δύναμη. Για παράδειγμα, ο μεγαλύτερος πρώτος αριθμός που είναι μικρότερος από το 2^{16} είναι ο 65521.



Σχήμα 10.2: Μετασχηματισμοί κλειδιού στον επεκτατό κατακερματισμό.

Στη συνέχεια, η τιμή του κλειδιού μετατρέπεται στον ισοδύναμο δυαδικό αριθμό και λαμβάνονται τα πρώτα ψηφία του, δηλαδή κάποια bits. Είναι δυνατόν επίσης να μη ληφθούν τα πρώτα bits αλλά τα τελευταία ή κάποια μεσαία, όπως συμβαίνει σε πολλές παραλλαγές της μεθόδου που συναντώνται στη βιβλιογραφία. Τα bits αυτά λαμβάνονται ως είσοδος σε κατάλογο, που περιέχει δείκτες προς τους κάδους του αρχείου. Ο κατάλογος αυτός είναι ένας μονοδιάστατος πίνακας με 2^d στοιχεία, όπου d είναι ο αριθμός των ψηφίων που επιλέγονται από το αποτέλεσμα της συνάρτησης κατακερματισμού και λέγεται **βάθος** (depth) ή **επίπεδο** (level) του καταλόγου.

Ο αριθμός των ψηφίων που εξάγονται από το αποτέλεσμα της συνάρτησης κατακερματισμού μεταβάλλεται χρονικά ανάλογα με τη μεταβολή του μεγέθους του αρχείου. Για το λόγο αυτό, το επίπεδο του καταλόγου είναι μία απαραίτητη παράμετρος. Ένα παράδειγμα παρουσιάζεται στο Σχήμα 10.3, όπου έχουν χρησιμοποιηθεί τα πρώτα τρία ψηφία του μετασχηματισμού, και συνεπώς ο πίνακας αποτελείται από οκτώ δείκτες. Αυτοί οι οκτώ δείκτες μπορούν να αναφερθούν σε οκτώ το πολύ κάδους. Έστω, λοιπόν, ότι η δυαδική μορφή του μετασχηματισμού ενός κλειδιού είναι



Σχήμα 10.3: Κατάλογος επιπέδου 3 με τέσσερις κάδους.

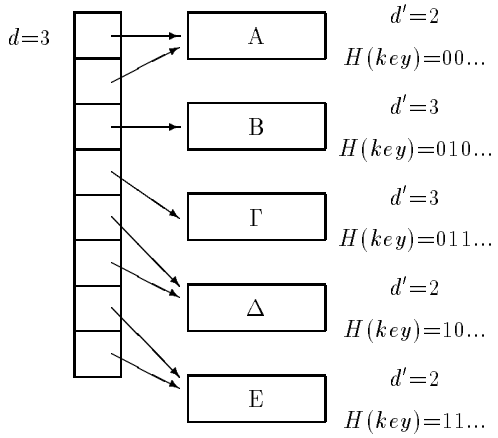
0110100101100101. Από αυτόν τον αριθμό απομονώνονται τα τρία πρώτα bits που ισοδυναμούν με το δεκαδικό αριθμό τρία. Άρα, με προσπέλαση στην υπ΄ αριθμό 3 είσοδο του πίνακα (που ουσιαστικά είναι η τέταρτη είσοδος του πίνακα) βρίσκεται ο κατάλληλος δείκτης που αναφέρεται στον κάδο Γ. Επίσης ας προσεχθεί ότι κάθε κάδος συνοδεύεται από μία παράμετρο, την d', που λέγεται **βάθος** (depth) ή **επίπεδο** (level) του κάδου. Η παράμετρος αυτή δηλώνει τον αριθμό των bits που είναι κοινός για τα κλειδιά όλων των εγγραφών του κάδου και είναι πάντα μικρότερη ή ίση από το βάθος του καταλόγου d. Συνεπώς, ο αριθμός των δεικτών που αναφέρονται σε ένα δεδομένο κάδο είναι $2^{d-d'}$.

Ανακεφαλαιώνοντας, η αναζήτηση στον επεκτατό κατακερματισμό γίνεται με πέντε βήματα. Πρώτον, εφαρμόζεται μία συνάρτηση κατακερματισμού στο κλειδί. Δεύτερον, εξάγονται τα πρώτα d bits. Τρίτο, χρησιμοποιείται ο κατάλογος για να εντοπισθεί ο κατάλληλος δείκτης. Τέταρτο, με βάση το δείκτη γίνεται προσπέλαση στον κάδο. Πέμπτο, γίνεται αναζήτηση μέσα στον κάδο για τον εντοπισμό του κλειδιού.

Η επεξεργασία των εισαγωγών και των διαγραφών εγγραφών είναι αρκετά πολύπλοκη. Ένας βασικός κανόνας είναι ότι η περιεκτικότητα ενός κάδου δεν μπορεί να είναι μεγαλύτερη από 100%, και δεν πρέπει να είναι μικρότερη από ένα ποσοστό που ορίζεται από το χρήστη, για παράδειγμα 50%. Όταν αυτός ο κανόνας παραβιάζεται τότε αλλάζει η δομή του αρχείου.

Έστω ότι στον κάδο Δ του αρχείου του Σχήματος 10.3 πρόκειται να

εισαχθεί μία εγγραφή. Αν ο κάδος Δ είναι ήδη πλήρης, τότε η νέα εγγραφή δεν μπορεί να αποθηκευθεί εκεί και προκαλείται επέκταση του αρχείου. Έτσι, ένας νέος κάδος Ε προστίθεται στο αρχείο. Οι μισοί από τους δείκτες που πριν αναφέρονταν στον κάδο Δ, τώρα αλλάζουν και αναφέρονται στον κάδο Ε. Οι αντίστοιχες εγγραφές μεταφέρονται από τον κάδο Δ στον κάδο Ε. Έτσι τελικά, οι δύο κάδοι Δ και Ε έχουν περιεκτικότητα 50% και υπάρχει αρκετός χώρος για μελλοντικές εισαγωγές. Η τελική μορφή του αρχείου φαίνεται στο Σχήμα 10.4. Ας προσεχθεί ότι πριν τη διάσπαση στον κάδο Δ βρίσκοταν όλες οι εγγραφές των οποίων ο μετασχηματισμός του κλειδιού άρχιζε από 1... Μετά τη διάσπαση στον κάδο Δ και στον κάδο Ε αποθηκεύονται όλες οι εγγραφές των οποίων ο μετασχηματισμός του κλειδιού αρχίζει από 10... και 11... αντίστοιχα.

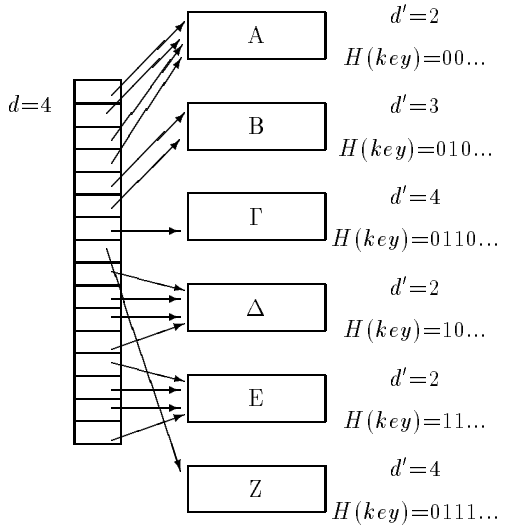


Σχήμα 10.4: Αναδιανομή κλειδιών λόγω διάσπασης κάδου.

Η διάσπαση των κάδων μπορεί να συνεχισθεί με τις διαδοχικές εισαγωγές νέων εγγραφών μέχρι ενός ορισμένου σημείου. Όταν ο κάδος που πρόκειται να διασπασθεί αναφέρεται από ένα μόνο δείκτη, δηλαδή $d = d'$, τότε ο κατάλογος πρέπει να επεκταθεί. Έστω ότι στον ήδη πλήρη κάδο Γ του αρχείου του Σχήματος 10.4 εισάγεται μία νέα εγγραφή. Τότε ένας νέος κάδος Ζ αποδίδεται στο αρχείο. Ωστόσο, ο κατάλογος δεν διαθέτει χώρο για να αποθηκευθούν οι σχετικοί δείκτες. Άρα, στο σημείο αυτό πρέπει να μεγεθυνθεί ο κατάλογος.

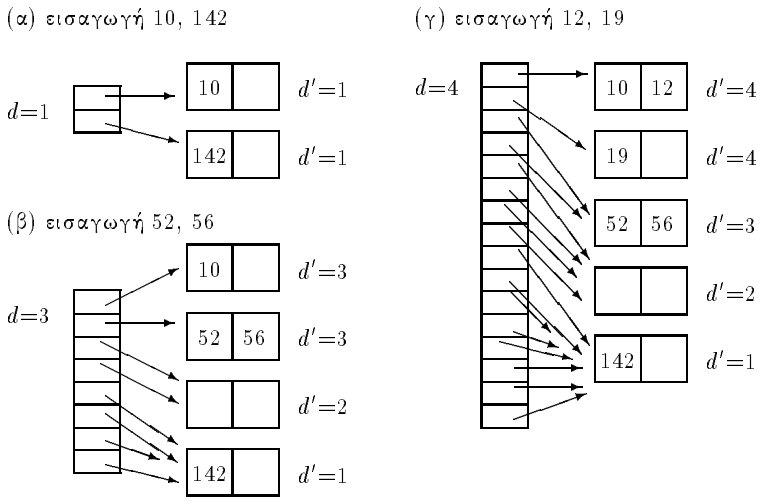
Σε κάθε διαδοχική μεγέθυνση ο κατάλογος διπλασιάζεται με αύξηση της παραμέτρου d κατά μία μονάδα. Κάθε δείκτης του αρχικού καταλόγου καταλαμβάνει δύο θέσεις του νέου καταλόγου. Για παράδειγμα, ο κατάλογος του

Σχήματος 10.4 περιέχει δείκτες που αναφέρονται κατά σειρά στους κάδους Α, Α, Β, Γ, Δ, Δ, Ε και Ε. Μετά την επέκταση του καταλόγου οι δείκτες αναφέρονται κατά σειρά στους κάδους Α, Α, Α, Α, Β, Β, Γ, Γ, Δ, Δ, Δ, Δ, Ε, Ε, Ε και Ε. Με το διπλασιασμό του καταλόγου ο κάδος Γ αναφέρεται πλέον από δύο δείκτες και συνεπώς μπορεί να προχωρήσει η διάσπαση. Έτσι, ο ένας από αυτούς τους δύο δείκτες στο εξής αναφέρεται στον κάδο Ζ, ενώ παράλληλα οι εγγραφές μοιράζονται στους δύο κάδους ανάλογα με τα πρώτα τέσσερα ψηφία από το αποτέλεσμα της συνάρτησης κατακερματισμού. Η νέα μορφή του αρχείου παρουσιάζεται στο Σχήμα 10.5.



Σχήμα 10.5: Διπλασιασμός καταλόγου.

Ο κατάλογος διπλασιάζεται κάθε φορά που διασπάται ένας κάδος, για τον οποίο ισχύει η σχέση $d = d'$, ανεξάρτητα από κατάσταση την άλλων κάδων. Αυτή η τεχνική μπορεί να οδηγήσει σε παθολογικές καταστάσεις. Για παράδειγμα, στο Σχήμα 10.6 παρουσιάζεται ένα αρχείο με δύο κάδους χωρητικότητας δύο εγγραφών, όπου αρχικά εισάγονται δύο εγγραφές με κλειδιά 10 ($= 00001010_2$) και 142 ($= 10001110_2$). Στη συνέχεια εισάγονται οι εγγραφές με κλειδιά 52 ($= 00110100_2$) και 56 ($= 00111000_2$), οπότε γίνονται δύο διαδοχικοί διπλασιασμοί του μεγέθους του καταλόγου, επειδή τα κλειδιά 10, 52 και 56 έχουν κοινά τα πρώτα δύο bits. Διπλασιασμός του καταλόγου γίνεται και με την εισαγωγή των εγγραφών με κλειδιά 12 ($= 00001100_2$) και 19 ($= 00010011_2$). Έτσι ο κατάλογος έχει $2^4 = 16$



Σχήμα 10.6: Παθολογική περίπτωση διπλασιασμού καταλόγων.

εισόδους, οπότε υπάρχουν αρκετοί διακριτοί δείκτες που δείχνουν είτε σε ένα κενό κάδο είτε σε ένα χοινό κάδο.

Εξ αιτίας διαδοχικών διαγραφών εγγραφών μπορεί να προκληθεί η αντίστροφη δυαδική διαδικασία της συρρίκνωσης του αρχείου. Δύο κάδοι συσσωματώνονται σε έναν κάδο κάτω από τρεις προϋποθέσεις:

- η μέση περιεκτικότητα των δύο κάδων δεν ξεπερνά το 50%, γιατί σε αντίθετη περίπτωση, δεν θα υπήρχε χώρος σε έναν κάδο για όλες τις εγγραφές,
- οι κάδοι που πρόκειται να συνδυασθούν χαρακτηρίζονται από την ίδια τιμή της παραμέτρου d', και
- τα κλειδιά των εγγραφών των δύο κάδων έχουν κοινά τα πρώτα $(d'-1)$ ψηφία του αποτελέσματος του μετασχηματισμού κατακερματισμού.

Οι δύο τελευταίες προϋποθέσεις είναι απαραίτητες, έτσι ώστε όλες οι εγγραφές του νέου κάδου να έχουν κοινά τα πρώτα d' ψηφία. Έστω για παράδειγμα, ότι η πρώτη συνθήκη συντρέχει για τους κάδους Α και Β του Σχήματος 10.5. Οι κάδοι αυτοί, όμως, δεν μπορούν να συγχωνευθούν γιατί τα κλειδιά των εγγραφών δεν έχουν τον ίδιο αριθμό κοινών ψηφίων. Απεναντίας, οι κάδοι Δ και Ε μπορούν να συγχωνευθούν γιατί ικανοποιούνται όλες οι συνθήκες.

Κατά τον ίδιο τρόπο, όταν όλοι οι δείκτες είναι κατά ζεύγη μπορεί να υποδιπλασιασθεί και το μέγεθος του καταλόγου. Για παράδειγμα, αν υποτεθεί ότι οι δείκτες του καταλόγου αναφέρονται κατά σειρά στους κάδους Α, Α, Α, Α, Β, Β, Γ, Γ, Δ, Δ, Δ, Δ, Ε, Ε, Ζ και Ζ, τότε με τη συρρίκνωση του καταλόγου οι δείκτες θα αναφέρονται στους κάδους Α, Α, Β, Γ, Δ, Δ, Ε και Ζ. Αντίθετα, αν οι δείκτες αφορούσαν στους κάδους Α, Α, Α, Β, Β, Β, Γ, Γ, Δ, Δ, Δ, Δ, Ε, Ε, Ζ και Ζ δεν θα μπορούσε να γίνει υποδιπλασιασμός του καταλόγου.

Ας σημειωθεί ότι με μιά άλλη απλή υλοποίηση του καταλόγου μπορεί να μην υπάρχουν περιττοί δείκτες. Δηλαδή, αν για δεδομένο επίπεδο d του καταλόγου και για κάποιο συνδυασμό d ψηφίων δεν υπάρχουν αντίστοιχες τιμές κλειδιών, τότε ο αντίστοιχος δείκτης έχει τιμή NIL. Η παραλλαγή αυτή έχει ως αποτέλεσμα ταχύτερη ανεπιτυχή αναζήτηση και απλούστερη διαδικασία διάσπασης κάδων. Ταυτόχρονα όμως έχει μεγαλύτερες απαιτήσεις χώρου γιατί θα χρειάζονται πολλοί κάδοι με λίγες εγγραφές ο καθένας και επομένως όσο μεγαλύτερος ο κάδος τόσο περισσότερος και ο μη χρησιμοποιημένος χώρος.

Θεωρητικά, είναι πιθανό όταν η περιεκτικότητα ενός κάδου πέσει κάτω από 50%, να μην υπάρχει κάποιος άλλος κάδος κατάλληλος ώστε οι δύο τους να συγχωνευθούν σε ένα νέο κάδο. Δηλαδή, δεν υπάρχει εγγύηση ότι η περιεκτικότητα κάθε κάδου και του αρχείου συνολικά είναι τουλάχιστον 50%. Αν

- ο αριθμός των εγγραφών, n, είναι μεγάλος,
- η χωρητικότητα των κάδων είναι επίσης μεγάλη, και
- τα κλειδιά διαμοιράζονται ομοιόμορφα μεταξύ των κάδων,

τότε αποδεικνύεται ότι είναι μηδαμινή η πιθανότητα να πέσει κάτω από 50% η περιεκτικότητα ενός κάδου. Η τελευταία υπόθεση σημαίνει ότι η χωρητικότητα των κάδων είναι περίπου η ίδια, άρα όλοι οι κάδοι διασπώνται περίπου ταυτόχρονα. Σε κάποια χρονική στιγμή, λοιπόν, κάποιοι κάδοι θα έχουν περιεκτικότητα περί το 100%, ενώ άλλοι θα έχουν περιεκτικότητα περί το 50%.

Η μέση περιεκτικότητα των κάδων είναι περίπου 69%, άρα η μέση τιμή του αριθμού των κάδων του αρχείου είναι $\frac{n}{Bkfr \times \ln 2}$. Η μέση τιμή του αριθμού των εισόδων του καταλόγου είναι $\frac{n}{Bkfr \times (\ln 2)^2}$. Από την τελευταία σχέση

φαίνεται ότι όταν το αρχείο διογκωθεί αρκετά, αντίστοιχα διογκώνεται και ο κατάλογος. Σε εξαιρετικές περιπτώσεις ο κατάλογος μπορεί να γίνει τόσο μεγάλος, ώστε να μην χωρά στην κύρια μνήμη, οπότε θα πρέπει να αποθηκευθεί στη δευτερεύουσα μνήμη και θα απαιτούνται επιπλέον προσπελάσεις στο δίσκο για επεξεργασία του πίνακα. Αυτό αποτελεί ένα σοβαρό μειονέκτημα της μεθόδου. Αυτό το πρόβλημα επιλύεται σε μία παραλλαγή της μεθόδου που λέγεται εκθετικός κατακερματισμός με περιορισμένο κατάλογο και παρουσιάζεται στη συνέχεια. Ένα άλλο μειονέκτημα της μεθόδου είναι ότι δεν προσφέρεται για ερωτήσεις διαστήματος, όπως για παράδειγμα η δομή του B^+-δένδρου.

10.4 Εκθετικός κατακερματισμός με περιορισμένο κατάλογο

Βασικό μειονέκτημα της προηγούμενης μεθόδου είναι ο ανεξέλεγκτος διπλασιασμός του μεγέθους του καταλόγου. Ένα άλλο μειονέκτημα της μεθόδου είναι ότι οι κάδοι διασπώνται περίπου ταυτόχρονα, επειδή οι εγγραφές κατανέμονται ισοπίθανα στους κάδους. Επομένως ενώ το κόστος της εισαγωγής εγγραφών διατηρείται περίπου σταθερό, περιοδικά το κόστος αυτό έχει εξάρσεις που οφείλονται στις πολλές και ταυτόχρονες διασπάσεις των κάδων του αρχείου και του καταλόγου. Η οργάνωση του εκθετικού κατακερματισμού με περιορισμένο κατάλογο, που προτάθηκε από το Lomet (1983), επιλύει και τα δύο προβλήματα.

Όσον αφορά στο πρώτο μειονέκτημα, ο κατάλογος δεν μπορεί να ξεπεράσει ένα μέγιστο μέγεθος που καθορίζεται από τη διαθέσιμη κύρια μνήμη. Δηλαδή, ο αριθμός των σελίδων που συμπεριλαμβάνονται σε έναν κατάλογο είναι δύναμη του δύο και άρα διαδοχικά διπλασιάζεται μέχρι του σημείου να φθάσει στο όριο των 2^{max} σελίδων. Για παράδειγμα, για μεγάλα αρχεία το μέγεθος αυτό θα μπορούσε να είναι 64, 128 κοκ. σελίδες. Έτσι ο κατάλογος διπλασιάζεται μέχρι να φθάσει αυτό το σταθερό οριακό μέγεθος και επομένως παραμένει σταθερός και ο αριθμός των κάδων που δεικτοδοτούνται από τον κατάλογο. Έτσι, μετά από αυτό το σημείο το αρχείο μεγαλώνει αυξάνοντας όχι τον αριθμό των κάδων αλλά το μέγεθος τους.

Όσον αφορά στο δεύτερο μειονέκτημα, ο μετασχηματισμός του κλειδιού σε τέτοια αρχεία ακολουθεί έναν αριθμό βημάτων. Αρχικά, τα κλειδιά

μετασχηματίζονται με βάση μία συνάρτηση ώστε να προκύψει μία ομοιόμορφη κατανομή. Υστερα, το αποτέλεσμα της συνάρτησης ανακατανέμεται εκθετικά με βάση τον τύπο:

$$exhash(k) \;=\; 2^{h(k)} \;-\; 1$$

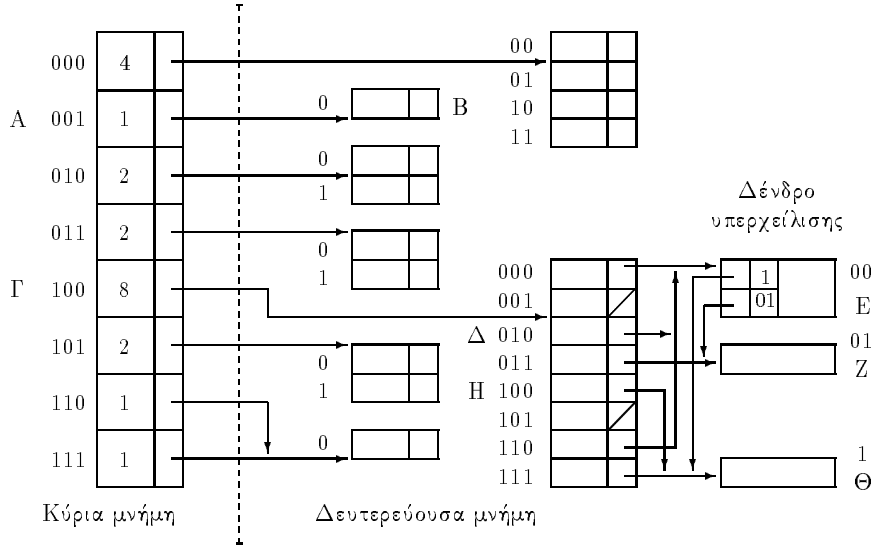
όπου k είναι το κλειδί, $h(k)$ είναι το αποτέλεσμα του πρώτου μετασχηματισμού, και $exhash(k)$ είναι το αποτέλεσμα του δεύτερου. Τελικό αποτέλεσμα είναι η ομοιόμορφη κατανομή να αντικατασταθεί με μία άλλη που διακρίνεται για τη συσσώρευση των κλειδιών προς το κάτω άκρο του διαστήματος των διευθύνσεων. Έτσι οι κάδοι δεν δέχονται τον ίδιο αριθμό εγγραφών και η επίδοση της εισαγωγής είναι περίπου σταθερή σε όλη τη διάρκεια της ζωής του αρχείου. Στον Πίνακα 10.2 παρατηρούμε ότι η απόσταση μεταξύ διαδοχικών τιμών της $exhash(k)$ μεγαλώνει καθώς οι τιμές της $h(k)$ αυξάνονται. Για παράδειγμα, αν και η απόσταση μεταξύ διαδοχικών τιμών της $h(k)$ είναι 0,067, εν τούτοις η απόσταση των πρώτων δύο τιμών της $exhash(k)$ είναι 0,47, ενώ η απόσταση των τελευταίων τιμών της $exhash(k)$ είναι 0,90.

$h(k)$	$exhash(k)$	$h(k)$	$exhash(k)$	$h(k)$	$exhash(k)$
0,000	0,000	0,400	0,320	0,800	0,741
0,067	0,047	0,467	0,382	0,867	0,823
0,133	0,097	0,533	0,447	0,933	0,910
0,200	0,149	0,600	0,516	1,000	1,000
0,267	0,203	0,667	0,662		
0,333	0,260	0,733	0,662		

Πίνακας 10.2: Εφαρμογή εκθετικού μετασχηματισμού.

Τα πρώτα bits του $exhash(k)$ καθορίζουν τη σελίδα του καταλόγου, όπου είναι αποθηκευμένος ο δείκτης προς τον κατάλληλο κάδο. Η επιλογή της σελίδας (από το σύνολο των σελίδων του κάδου), όπου θα πρέπει να συνεχισθεί η αναζήτηση, γίνεται με βάση μερικά από τα επόμενα bits από το $exhash(k)$. Ακόμη σε κάθε κάδο αντιστοιχεί μία περιοχή υπερχείλισης, που είναι οργανωμένη ως δυαδικό δένδρο που ονομάζεται **ο-δένδρο** (από το overflow). Ο κατάλογος για αυτό το δένδρο αποθηκεύεται στην πρώτη σελίδα του αντίστοιχου κάδου. Κατά παρόμοιο τρόπο, η διαχείριση της περιοχής υπερχείλισης γίνεται με τα επόμενα bits του $exhash(k)$.

Οι προηγούμενες έννοιες διασαφηνίζονται με το παράδειγμα του Σχήματος 10.7. Αριστερά φαίνεται ο κατάλογος που έχει φθάσει στα όρια της



Σχήμα 10.7: Εκθετικός κατακερματισμός με περιορισμένο κατάλογο.

ανάπτυξής του, επειδή max=3. Ο κατάλογος περιέχει δείκτες προς τους κύριους κάδους. Οι δύο τελευταίοι δείκτες αναφέρονται στον ίδιο κάδο, γεγονός που άλλωστε είναι δυνατό και στη μέθοδο του απλού επεκτατού κατακερματισμού.

Ας υποτεθεί ότι πρέπει να αναζητηθεί το κλειδί με τιμή 204. Έστω ότι $exhash(204) = (.0011001100_2)$. Επειδή ο κατάλογος αποτελείται από οκτώ σελίδες, θεωρούνται τα τρία πρώτα bits για την προσπέλαση του καταλόγου, δηλαδή τα 001. Άρα η αναζήτηση κατευθύνεται στη δεύτερη σελίδα του καταλόγου, τη σελίδα Α. Επειδή η είσοδος του μεγέθους στον κατάλογο είναι 1, ο αντίστοιχος δείκτης αναφέρεται σε κάδο με μία μόνο σελίδα. Άρα τελικά, η αναζήτηση τερματίζεται στον κάδο Β του Σχήματος 10.7.

Στη συνέχεια, ας υποτεθεί ότι πρέπει να αναζητηθεί το κλειδί 641. Κατά τον ίδιο τρόπο, έστω ότι ισχύει $exhash(641) = (.1000101100_2)$. Τα τρία πρώτα bits είναι τα 100, άρα η αναζήτηση κατευθύνεται στην πέμπτη σελίδα του καταλόγου, τη σελίδα Γ. Επειδή η είσοδος του μεγέθους στον κατάλογο είναι 3, ο αντίστοιχος δείκτης αναφέρεται σε κάδο με οκτώ σελίδες και συνεπώς πρέπει να χρησιμοποιηθούν τα τρία επόμενα bits για τη διευκρίνηση της συγκεκριμένης σελίδας, όπου πρέπει να συνεχισθεί η αναζήτηση. Αυτά τα bits είναι 010 και συνεπώς η αναζήτηση συνεχίζεται στην τρίτη σελίδα,

τη σελίδα Δ. Αν η σελίδα Δ δεν περιέχει το κλειδί, τότε η αναζήτηση πρέπει να συνεχισθεί στο ο-δένδρο. Στη σελίδα Δ υπάρχει δείκτης που αναφέρεται στην πρώτη σελίδα του δένδρου, τη σελίδα Ε του καταλόγου του δένδρου. Αν το τέταρτο και το πέμπτο bit είναι 00 ή 01, τότε το κλειδί περιέχεται στη σελίδα Ε ή στη σελίδα Ζ, αντίστοιχα. Ειδάλλως, αν το τέταρτο bit είναι 1, τότε η αναζήτηση συνεχίζεται στη σελίδα Θ. Έτσι, στην περίπτωση αναζήτησης του κλειδιού 641 απαιτούνται τρεις προσπελάσεις στο δίσκο, δηλαδή διαδοχικά στις σελίδες Δ, Ε και Θ. Αν παρουσιασθεί το φαινόμενο να χρειασθούν περισσότερο από δύο προσπελάσεις, τότε οι δείκτες αλλάζουν ώστε να συντομευθεί η διαδικασία. Έτσι ο δείκτης της σελίδας Δ δεν θα δείχνει πλέον στη σελίδα Ε, αλλά στη σελίδα Ζ.

Τέλος, ας υποτεθεί ότι πρέπει να αναζητηθεί το κλειδί 670. Κατά τον ίδιο τρόπο, ισχύει $exhash(670) = (.1001001011_2)$. Τα τρία πρώτα bits είναι τα 100, άρα η αναζήτηση και πάλι κατευθύνεται στη σελίδα Γ. Τα τρία επόμενα bits είναι 100 και συνεπώς η αναζήτηση συνεχίζεται στην πέμπτη σελίδα που ονομάζεται Η. Αν το κλειδί δεν είναι αποθηκευμένο στη σελίδα Η, τότε η αναζήτηση πρέπει να συνεχισθεί στο ο-δένδρο. Στη σελίδα Η υπάρχει δείκτης που αναφέρεται στη σελίδα Θ, που είναι ένα φύλλο του ο-δένδρου. Δηλαδή, τελικά το κλειδί 670 βρίσκεται ή στη σελίδα Η ή στη σελίδα Θ.

Όπως αναφέρθηκε όταν ένας κάδος γεμίσει, τότε το μέγεθος του διπλασιάζεται και ενημερώνεται κατάλληλα η αντίστοιχη είσοδος του καταλόγου. Ο κάδος μεταφέρεται από το συγκεκριμένο σημείο του δίσκου και ξαναγράφεται σε κάποια άλλη περιοχή του δίσκου ώστε να μην χρειάζονται δύο προσπελάσεις. Μία άλλη συνηθισμένη τεχνική είναι να αφήνονται εγγραφές στην υπερχείλιση παρά να διπλασιάζεται συνεχώς το μέγεθος του κάδου, οπότε με αυξημένο κόστος αναζήτησης επιτυγχάνεται καλύτερη χρήση του χώρου. Σε σχέση με την απλή μέθοδο του επεκτατού κατακερματισμού υπάρχουν τα εξής πλεονεκτήματα: δεν χρειάζεται προσπέλαση στο δίσκο για τον κατάλογο, ενώ απαιτείται μόνο μία προσπέλαση στο δίσκο αν δεν υπάρχει υπερχείλιση.

10.5 Γραμμικός κατακερματισμός

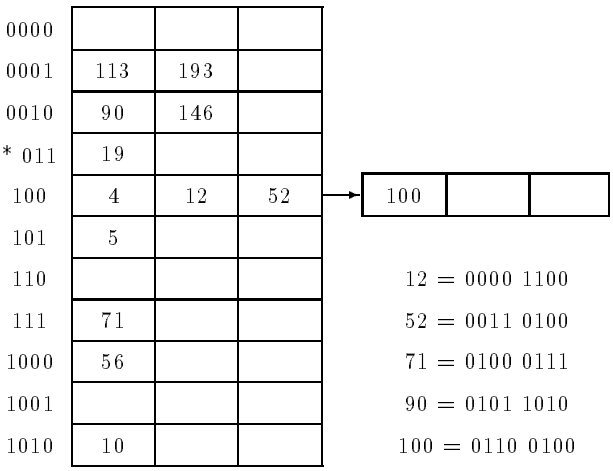
Η μέθοδος αυτή προτάθηκε από τον Litwin (1980) και γνώρισε πολλές βελτιώσεις με παραλλαγές. Αν και είναι διαφορετική από τις προηγούμενες γιατί δεν έχει κατάλογο, έχει ωστόσο πολλές ομοιότητες με αυτές. Στο

κλειδί μίας εγγραφής εφαρμόζεται μία συνάρτηση κατακερματισμού (κατά τα γνωστά) και απομονώνονται τα τελευταία (αντί τα πρώτα) k δυαδικά ψηφία του αποτελέσματος. Για παράδειγμα, αν αρχικά το αρχείο έχει οκτώ κάδους, τότε απομονώνονται τα τρία τελευταία bits του αποτελέσματος της συνάρτησης του κατακερματισμού για τον εντοπισμό του κατάλληλου κάδου. Στο Σχήμα 10.8 παρουσιάζεται αυτή η περίπτωση, όπου κάθε κάδος έχει χωρητικότητα τρεις εγγραφές.

000	56			4 = 0000 0100
001	113	193		5 = 0000 0101
010	10	146		10 = 0000 1010
011	19			19 = 0001 0011
100	4			56 = 0011 1000
101	5			113 = 0111 0001
110				146 = 1001 0010
111				193 = 1100 0001

Σχήμα 10.8: Γραμμικός κατακερματισμός με χρήση τριών bits.

Το αρχείο αυξάνει με διαδοχικές διασπάσεις των κάδων, όπου οι εγγραφές ενός κάδου που έχουν κοινά τα τελευταία k bits, μοιράζονται σε δύο κάδους ανάλογα με την τιμή των τελευταίων $k+1$ bits. Στο Σχήμα 10.8 οι εγγραφές με κλειδιά 113 και 193 είναι αποθηκευμένες στον κάδο, που χαρακτηρίζεται από τα τρία ψηφία 001. Η διανομή αυτών των εγγραφών σε δύο κάδους γίνεται ανάλογα με την τιμή των τεσσάρων τελευταίων bits. Κατά σύμπτωση και οι δύο εγγραφές θα αποθηκευθούν και πάλι στον ίδιο κάδο. Δεν συμβαίνει όμως το ίδιο στην περίπτωση των εγγραφών με κλειδιά 10 και 146, που τελικά αποθηκεύονται σε διαφορετικούς κάδους. Αυτό φαίνεται στο Σχήμα 10.8 όπου έχουν διασπασθεί οι τρεις πρώτοι κάδοι. Έτσι, όμως μερικοί κάδοι περιέχουν εγγραφές με κλειδιά που έχουν k κοινά bits, ενώ άλλοι περιέχουν εγγραφές με κλειδιά που έχουν $k+1$ κοινά bits. **Οριακή τιμή** (boundary value) ονομάζεται η τιμή πέρα από την οποία οι κάδοι διακρίνονται από τα τελευταία k bits. Στην επικεφαλίδα κάθε αρχείου γραμμικού κατακερματισμού αποθηκεύεται η οριακή τιμή και η αντίστοιχη τιμή του k. Στο Σχήμα 10.9 οι τιμές αυτές είναι 011 και 3 αντίστοιχα.



Σχήμα 10.9: Γραμμικός κατακερματισμός με χρήση τριών και τεσσάρων bits.

10.5.1 Προσπέλαση εγγραφής

Κατά την αναζήτηση εγγραφής απομονώνονται τα τελευταία k bits από το αποτέλεσμα της συνάρτησης κατακερματισμού. Αν η τιμή που προκύπτει με χρήση των k bits είναι μικρότερη από την οριακή τιμή, τότε χρησιμοποιούνται $(k+1)$ bits. Έτσι, αν στο αρχείο του παραδείγματος του Σχήματος 10.11 αναζητείται μία εγγραφή με τελευταία bits τα 1000 ή τα 1101, τότε χρησιμοποιούνται 4 ή 3 bits από αυτά, αντίστοιχα.

Η επίδοση της αναζήτησης σε γραμμικό αρχείο είναι πολύ καλή και προσεγγίζει τη μία προσπέλαση στο δίσκο. Αν η χωρητικότητα του κάδου, $Bkfr$, είναι 50, ενώ ο παράγοντας φόρτισης, Lf, είναι 75%, τότε η επιτυχής και η ανεπιτυχής αναζήτηση απαιτούν 1.05 και 1.27 προσπελάσεις, αντίστοιχα. Κάτω από αυτές τις συνθήκες το χρονικό κόστος για την αναζήτηση είναι:

$$\begin{aligned} T_{\text{προσ}}(\text{επιτ}) &= 1.05 \times (s + r + dtt) \\ T_{\text{προσ}}(\text{ανεπ}) &= 1.27 \times (s + r + dtt) \end{aligned}$$

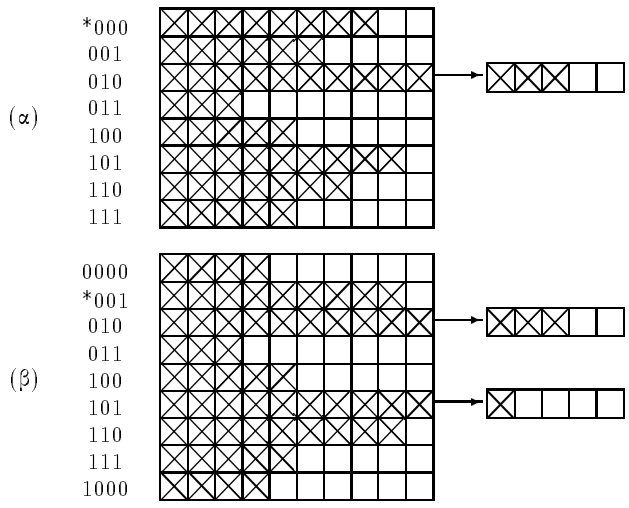
Ο αριθμός των προσπελάσεων είναι συνήθως μεγαλύτερος από τον αντίστοιχο αριθμό του στατικού αρχείου κατακερματισμού με χρήση αλυσίδων,

όταν ο παράγοντας φόρτισης είναι ίδιος. Αυτό οφείλεται στο ότι οι εγγραφές δεν κατανέμονται ομοιόμορφα μεταξύ των κάδων στο γραμμικό κατακερματισμό. Οι κάδοι που αναφέρονται με k bits δέχονται συνήθως περισσότερες εγγραφές απ΄ ότι οι κάδοι που αναφέρονται με $k+1$ bits. Ας προσεχθεί, όμως, ότι ο αριθμός των προσπελάσεων δεν εξαρτάται από τον αριθμό των εγγραφών στο αρχείο. Δηλαδή, ακόμη και αν στο αρχείο εισαχθούν πολλαπλάσιες εγγραφές η επίδοση θα παραμείνει σταθερή. Αυτό είναι το μεγάλο πλεονέκτημα της μεθόδου αυτής έναντι των άλλων μεθόδων οργάνωσης αρχείων και ιδιαίτερα έναντι της απλής τεχνικής κατακερματισμού με χρήση αλυσίδων.

10.5.2 Εισαγωγή εγγραφής

Η διαδικασία εισαγωγής σε αρχείο γραμμικού κατακερματισμού θα δοθεί με τη βοήθεια ενός παραδείγματος. Ας υποτεθεί ότι ένα αρχείο αποτελείται από οκτώ κύριους κάδους με χωρητικότητα δέκα εγγραφών. Έστω, επίσης, ότι ο παράγοντας φόρτισης δεν πρέπει να ξεπεράσει το 70%. Εκτός από την κύρια περιοχή υπάρχει και η περιοχή υπερχείλισης, όπου χρησιμοποιείται η τεχνική της αλυσίδας όπως ακριβώς και στην απλή στατική μέθοδο.

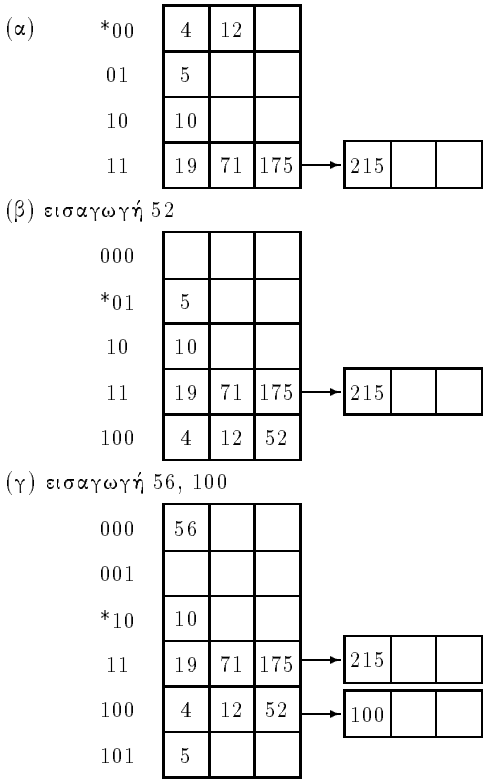
Στο Σχήμα 10.10α παρουσιάζεται αρχείο γραμμικού κατακερματισμού



Σχήμα 10.10: Επέκταση αρχείου γραμμικού κατακερματισμού.

που σε μία δεδομένη χρονική στιγμή περιέχει 56 εγγραφές. Επομένως, ο παράγοντας φόρτισης είναι ακριβώς 70%. Αν προστεθούν άλλες 7 εγγραφές τότε χρειάζεται ένας ακόμη κάδος, ώστε ο παράγοντας φόρτισης να παραμείνει στο 70%. Δηλαδή, γενικά όταν εισάγονται $Lf \times Bkfr$ εγγραφές, τότε το αρχείο αποκτά ένα κάδο ακόμη. Έτσι, ο πρώτος κάδος που χαρακτηρίζεται από το string 000 διασπάται και προκύπτουν δύο κάδοι με αντίστοιχα strings τα 0000 και 1000. Η διαχείριση των εισαγωγών γίνεται με βάση τις πληροφορίες που υπάρχουν αποθηκευμένες στην επικεφαλίδα του αρχείου, την οριακή τιμή και την τιμή του k. Το αποτέλεσμα της εισαγωγής παρουσιάζεται στο Σχήμα 10.10β.

Η διάσπαση των κάδων και η τακτοποίηση της υπερχείλισης είναι ανεξάρτητα γεγονότα. Δηλαδή, ούτε η υπερχείλιση προκαλεί απαραίτητα διάσπαση,



Σχήμα 10.11: Γραμμική διάσπαση κάδων.

αλλά ούτε και η διάσπαση μειώνει απαραίτητα την υπερχείλιση. Πιο συγκεκριμένα, οι διασπάσεις εκτελούνται γραμμικά από την αρχή προς το τέλος του αρχείου, άσχετα από τον κάδο που υπερχείλισε. Για παράδειγμα, η δομή του Σχήματος 10.11 αποτελείται από τέσσερις κάδους χωρητικότητας τριών εγγραφών. Αν ο παράγοντας φόρτισης δεν πρέπει να υπερβεί το 67%, τότε το πολύ οκτώ εγγραφές μπορεί να αποθηκευθούν στο αρχείο υπό αυτή τη μορφή. Αυτή η κατάσταση παρουσιάζεται στο Σχήμα 10.11α, όπου ο τέταρτος κάδος συνοδεύεται από αλυσίδα υπερχείλισης. Με την εισαγωγή της ένατης εγγραφής πρέπει να διασπασθεί ο πρώτος κάδος, όπως φαίνεται στο Σχήμα 10.11β. Με την εισαγωγή τριών ακόμη εγγραφών πρέπει να διασπασθεί ο δεύτερος κάδος. Η νέα κατάσταση παρουσιάζεται στο Σχήμα 10.11γ, όπου πλέον οι δύο κάδοι έχουν αλυσίδα υπερχείλισης. Ο τέταρτος κάδος θα διασπασθεί μετά την εισαγωγή τεσσάρων ακόμη εγγραφών.

Πειραματικά αποτελέσματα έχουν δείξει ότι σε αρχείο με $Bkfr$=50 και Lf=75% απαιτούνται κατά μέσο όρο 2,62 προσπελάσεις στο δίσκο για να εισαχθεί μία νέα εγγραφή. Αυτή η τιμή εξηγείται ως εξής. Κατ΄ αρχήν μία προσπέλαση απαιτείται για να έρθει στην κύρια μνήμη ο κύριος κάδος. Με πιθανότητα $1/Bkfr$ (για την ακρίβεια με ακόμη μεγαλύτερη πιθανότητα) θα δημιουργηθεί ένας νέος κάδος. Τότε θα πρέπει οι εγγραφές του αρχικού κάδου να μοιρασθούν στους δύο νέους κάδους. Αυτό σημαίνει ότι θα γίνουν άλλες δύο προσπελάσεις στο δίσκο για την επανεγγραφή τους. Επιπλέον, αν ο αρχικός κάδος διαθέτει και αλυσίδα υπερχείλισης, τότε θα είναι απαραίτητο να προσπελασθεί και αυτή. Τελικά, προκύπτει ότι το χρονικό κόστος για μία εισαγωγή είναι:

$$T_{\text{εισ}} \;=\; T_{\text{προσ}}(\text{επιτ}) \;+\; 2r \;+\; \frac{s+r+dtt}{Bkfr} \;+\; \frac{s+r+dtt+2r+s+r+dtt}{Lf \times Bkfr}$$

Αν $Bkfr$=50 και Lf=0.75, τότε το κόστος αυτό ισούται με:

$$\begin{aligned} T_{\text{εισ}} \;&=\; \left(1.27+\frac{1}{50}+\frac{2}{37}\right)\times(s+r+dtt) \;+\; \left(1+\frac{1}{37}\right)\times 2r \\ &=\; 74\text{ms} \;=\; 2.34\times 31.8\text{ms} \;=\; 2.34\times(s+r+dtt) \end{aligned}$$

που προσεγγίζει πολύ την πειραματική τιμή. Το αξιοπρόσεκτο είναι ότι το κόστος εισαγωγής μίας εγγραφής είναι λίγο περισσότερο από δύο προσπελάσεις στο δίσκο. Δηλαδή, φαίνεται ότι το κόστος αυτό είναι σχετικά μεγάλο, ιδιαίτερα αν συγκριθεί με το αντίστοιχο κόστος εισαγωγής σε στατικό αρχείο κατακερματισμού με αλυσίδες. Ωστόσο, τα φαινόμενα απατούν. Το

κόστος εισαγωγής εγγραφής σε αρχείο γραμμικού κατακερματισμού συμπεριλαμβάνει και το κόστος τοπικής αναδιοργάνωσης. Αντίθετα, στο στατικό αρχείο η αναδιοργάνωση γίνεται περιοδικά. Σε τελευταία ανάλυση, η εισαγωγή δεν είναι χρονοβόρα πράξη.

10.5.3 Διαγραφή εγγραφής

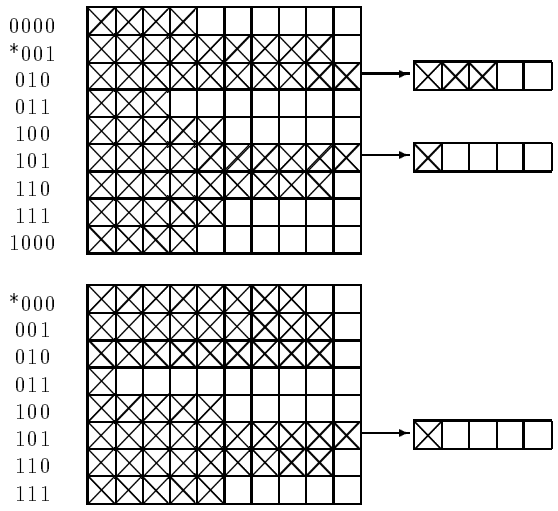
Η διαγραφή εγγραφής μπορεί να γίνει κατά πολλούς τρόπους. Μία λύση είναι να έρθουν στην κύρια μνήμη όλες οι εγγραφές του κάδου και της αλυσίδας υπερχείλισης και να μεταφερθεί η τελευταία της αλυσίδας στη θέση της εγγραφής που διαγράφεται. Αν η εγγραφή που πρόκειται να διαγραφεί είναι τελευταία στην αλυσίδα, τότε απλώς ελευθερώνεται ο χώρος. Αν ο αριθμός των εγγραφών του αρχείου είναι κατά $Bkfr \times Lf$ μικρότερος από αυτόν που υπαγορεύεται από το δεδομένο παράγοντα φόρτισης, τότε το αρχείο συρρικνώνεται κατά ένα κάδο.

Κάτω από αυτές τις συνθήκες το χρονικό κόστος για τη διαγραφή εγγραφής είναι:

$$T_{\delta\iota\alpha\gamma} \;=\; T_{\pi\rho o\sigma}(\alpha\nu\epsilon\pi) \;+\; 2r$$

Αυτός ο τύπος δεν συμπεριλαμβάνει το κόστος συρρίκνωσης του αρχείου. Η πιθανότητα για το γεγονός αυτό είναι πρακτικά μικρή αν υποτεθεί ότι ο ρυθμός άφιξης εγγραφών είναι πιο μεγάλος από το ρυθμό διαγραφών.

Από αλγοριθμική άποψη η διαδικασία συρρίκνωσης είναι η αντίστροφη της διαδικασίας διάσπασης λόγω εισαγωγής. Παρακολουθώντας το συνολικό αριθμό εγγραφών του αρχείου διαπιστώνεται πότε υπάρχουν $Bkfr \times Lf$ εγγραφές λιγότερο από το συγκεκριμένο παράγοντα φόρτισης. Τότε ο τελευταίος κάδος συγχωνεύεται με τον κάδο που έχει κοινά τα k τελευταία bits. Ένα παράδειγμα συρρίκνωσης λόγω διαγραφών φαίνεται στο Σχήμα 10.12, όπου το γραμμικό αρχείο αποτελείται από 9 κάδους με χωρητικότητα δέκα εγγραφές ανά κάδο. Ας υποτεθεί ότι διαγράφονται τέσσερις εγγραφές από τον κάδο 010, δύο εγγραφές από τον κάδο 011 και μία από τον κάδο 1000, οπότε απαιτείται συρρίκνωση του αρχείου. Από τους δύο κάδους 0000 και 1000 δημιουργείται ένας, ο κάδος 000 με τις εγγραφές των δύο προηγούμενων.



Σχήμα 10.12: Συρρίκνωση αρχείου γραμμικού κατακερματισμού.

10.5.4 Εξαντλητική ανάγνωση αρχείου

Όπως στη στατική έκδοση του αρχείου κατακερματισμού, έτσι και ο γραμμικός κατακερματισμός δεν διατηρεί την τάξη των κλειδιών των εγγραφών. Άρα, αν υπάρχει μία λίστα όλων των κλειδιών των εγγραφών, τότε ο χρόνος για την εξαντλητική ανάγνωση του αρχείου ισούται με:

$$T_{\text{εξαν}} \;=\; n \times 1.05 \times (s + r + dtt)$$

Αν δεν ενδιαφέρει να δοθούν στο χρήστη οι εγγραφές ταξινομημένες με βάση την τιμή κάποιου κλειδιού, τότε η διαδικασία επιταχύνεται και απλά σαρώνεται το αρχείο από την αρχή ως το τέλος. Ωστόσο, εξαιτίας του κενού χώρου στο τέλος των κάδων, η εξαντλητική αυτή ανάγνωση είναι πιο αργή από την αντίστοιχη σε ένα αρχείο σωρού.

Λόγω της μη ύπαρξης καταλόγου, ο γραμμικός κατακερματισμός είναι η πιο ενδιαφέρουσα τεχνική δυναμικών τυχαίων αρχείων και έχει γνωρίσει πολλές επεκτάσεις. Στη συνέχεια αναφέρονται επιγραμματικά μόνο δύο από αυτές. Σύμφωνα με το γραμμικό κατακερματισμό **με μερικές επεκτάσεις** (with partial expansions), που προτάθηκε από τον Larson (1980), όταν ο παράγοντας χρησιμοποίησης χώρου υπερβεί την προκαθορισμένη τιμή, τότε δεν διασπάται μόνο ένας κάδος αλλά οι εγγραφές ενός ζεύγους αναδιανέμονται μεταξύ τριών κάδων. Οι κάδοι ενός ζεύγους απέχουν $Bkfr/2$ κάδους

και επιλέγονται γραμμικά μέχρι να τελειώσουν, οπότε έχει συμβεί μία μερική διάσπαση. Στη συνέχεια με παρόμοια τεχνική επιλέγονται τριάδες κάδων μέχρι το πέρας της δεύτερης μερικής διάσπασης και την ολοκλήρωση μίας πλήρους επέκτασης. Η μέθοδος αυτή βελτιώνει την επίδοση της αναζήτησης και τη χρήση του χώρου με τίμημα το αυξημένο κόστος εισαγωγής σε σχέση με την αρχική μέθοδο. Μία άλλη επέκταση της αρχικής μεθόδου έγινε από τον Tharp (1987) και λέγεται γραμμικός κατακερματισμός με **προτεραιότητα διάσπασης** (priority splitting). Αν γίνει υπέρβαση της καθορισμένης τιμής του παράγοντα χρησιμοποίησης χώρου, τότε δίνεται προτεραιότητα στη διάσπαση του κάδου με τη μεγαλύτερη αλυσίδα υπερχείλισης. Βέβαια κάθε κάδος διασπάται μία μόνο φορά κατά τη διάρκεια μίας επέκτασης-διπλασιασμού του αρχείου ακόμη και αν συνεχώς δέχεται εγγραφές στην υπερχείλισή του. Για την υλοποίηση αυτής της μεθόδου απαιτούνται επιπλέον δομές που όμως είναι σχετικά μικρές και αποθηκεύονται στην κύρια μνήμη. Για όλες τις παραλλαγές του γραμμικού κατακερματισμού ισχύει η γενική παρατήρηση ότι η αναζήτηση γίνεται με μία περίπου προσπέλαση στο δίσκο. Όμως, η δομή αυτή δεν προσφέρεται για ερωτήσεις διαστήματος που απαντώνται πολύ αποτελεσματικά από τα B^+-δένδρα. Σε επόμενο κεφάλαιο θα περιγραφεί μέθοδος που εγγυάται την επιτυχή αναζήτηση με μία προσπέλαση.

10.6 Ασκήσεις

<1> Ποιός είναι ο ελάχιστος και ο μέγιστος αριθμός κάδων που απαιτούνται από αρχείο δυναμικού κατακερματισμού με $Bkfr$=2 για την αποθήκευση 10 εγγραφών, αν όλες κατευθύνονται στο ίδιο δυαδικό δένδρο του καταλόγου;

<**2**> Σε αρχείο δυναμικού κατακερματισμού με $Bkfr$=2 να εισαχθούν διαδοχικά οι εγγραφές με τιμές κλειδιών 42, 57, 16, 52, 66, 77, 12, 25, 21 και 33. Να θεωρηθούν οι συναρτήσεις $h_1(key)$, $h_2(key)$ καθώς και η ψευδοτυχαία γεννήτρια του Κεφαλαίου 10.2. Η άσκηση να επαναληφθεί εφαρμόζοντας την παραλλαγή της αναβολής διάσπασης με β=1,5.

<**3**> Να σχεδιασθεί η μορφή ενός αρχικά κενού αρχείου επεκτατού κατακερματισμού κατά την εισαγωγή διαδοχικά των κλειδιών 27, 18, 29, 28, 42, 13 και 16. Ως συνάρτηση κατακερματισμού να ληφθεί η $h(key) = key \bmod 11$, ώστε από τα κλειδιά αυτά να προκύψουν τετραψήφιοι δυαδικοί αριθμοί και

να θεωρηθεί κάδος χωρητικότητας τριών εγγραφών. Στη συνέχεια να διαγραφούν τα κλειδιά 16, 29 και 13. Η άσκηση να επιλυθεί θεωρώντας αρχικά τα πρώτα bits και κατόπιν τα τελευταία bits.

<4> Να επιλυθεί η πρώτη άσκηση θεωρώντας ότι οι δείκτες του καταλόγου μπορούν να πάρουν τιμές NIL.

<**5**> Ποιός είναι ο ελάχιστος αριθμός bits που απαιτείται σε αρχείο επεκτατού κατακερματισμού με

- 10.000 εγγραφές και χωρητικότητα 25 εγγραφές ανά κάδο,
- 100.000 εγγραφές και χωρητικότητα 10 εγγραφές ανά κάδο, και
- 1.000.000 εγγραφές και χωρητικότητα 8 εγγραφές ανά κάδο.

Ποιά είναι η πιθανότητα να μην υπάρχουν συνώνυμα σε αρχείο με 10.000 εγγραφές, όταν η συνάρτηση κατακερματισμού δίνει τιμές στο διάστημα 216, 224 και 232.

<**6**> Σε εκθετικό αρχείο με περιορισμένο κατάλογο ο κατάλογος αποτελείται από τέσσερις σελίδες με τέσσερα ζεύγη κλειδιών-δεικτών ανά σελίδα, ενώ κάθε σελίδα δεδομένων μπορεί να αποθηκεύσει δύο εγγραφές. Η συνάρτηση κατακερματισμού δίνει οκταψήφιους δυαδικούς αριθμούς. Τα πρώτα δύο ψηφία προσδιορίζουν τη σελίδα του καταλόγου, ενώ τα δύο επόμενα προσδιορίζουν τη διεύθυνση της σελίδας δεδομένων. Να σχεδιασθεί το αποτέλεσμα της εισαγωγής των εγγραφών με κλειδιά 255, 200, 56, 64, 155, 67, 43, 12, 205, 132, 128, 144 και 79.

<**7**> Σε αρχείο γραμμικού κατακερματισμού με $Bkfr$=2 και Lf=0.5 να εισαχθούν διαδοχικά οι εγγραφές με τιμές κλειδιών 42, 57, 16, 52, 66, 77, 12, 25, 21 και 33. Αρχικά το αρχείο αποτελείται από τέσσερις κάδους. Οι τιμές των κλειδιών να μετατραπούν σε επταψήφιους δυαδικούς αριθμούς και να απομονωθούν τα τελευταία ψηφία. Η άσκηση να επαναληφθεί για $Bkfr$=3 και Lf=0.67.

<**8**> Στο αρχείο της προηγούμενης άσκησης να εισαχθούν δέκα εγγραφές, ώστε να προκληθεί υπερχείλιση με αλυσίδα δύο κάδων.

<**9**> Σε αρχείο γραμμικού κατακερματισμού με $Bkfr$=3, Lf=0.67 και τέσσερις αρχικούς κάδους εισάγονται διαδοχικά οι επόμενες 24 εγγραφές με τιμές κλειδιών 42, 57, 16, 52, 66, 77, 12, 25, 21, 33, 32, 14, 50, 49, 47, 7, 56, 13, 62, 27, 31, 41, 30, 4, 10 και 6. Ποιό είναι τα αρχικό μέγεθος του

αρχείου; Πόσα ψηφία θα χρησιμοποιηθούν διαδοχικά; Ποιά θα είναι η δομή του αρχείου με τις διαδοχικές επεκτάσεις;

<10> Να επαναληφθεί η προηγούμενη άσκηση εφαρμόζοντας την παραλλαγή του γραμμικού κατακερματισμού με μερικές επεκτάσεις.

<11> Να επαναληφθεί η προηγούμενη άσκηση εφαρμόζοντας την παραλλαγή του γραμμικού κατακερματισμού με προτεραιότητα διάσπασης.