

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΜΑΘΗΜΑ
 ΑΚΑΔ. ΕΤΟΣ

ΒΑΣΕΙΣ ΔΕΛΟΜΕΝΩΝ
 2011-12

ΔΙΔΑΣΚΟΝΤΕΣ

Ιωάννης Βασιλείου Καθηγητής
 Τιμολέων Σελλής Καθηγητής

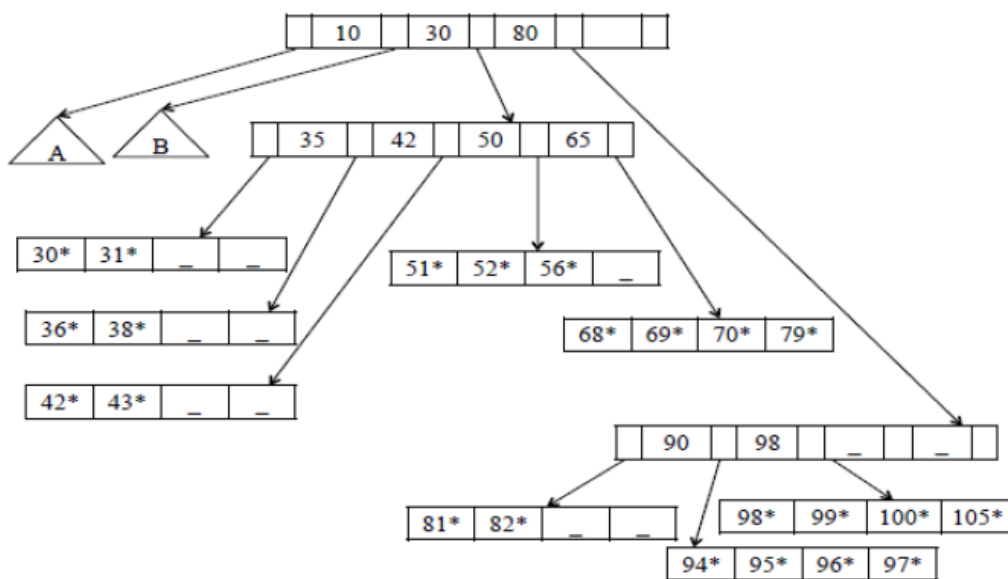
ΛΥΣΗ ΤΗΣ ΔΕΥΤΕΡΗΣ ΑΣΚΗΣΗΣ

Όλγα Γκουντούνα
 (olga@dblab.ece.ntua.gr)

Άσκηση 1 [25 μονάδες]

Θεωρείστε ότι θέλουμε να διαγράψουμε την τιμή **43** στο B+ δέντρο της Εικόνας 1. Η διαγραφή αυτή προκαλεί μείωση της πληρότητας του φύλλου που περιέχει τα 42*, 43*, κάτω του 50%. Θεωρείστε τις παρακάτω τρεις επιλογές:

- (1) Συγχώνευση του φύλλου με τον δεξί αδελφό του.
 - (2) Συγχώνευση του φύλλου με τον αριστερό αδελφό του.
 - (3) Ανακατανομή των εγγραφών του φύλλου με τον δεξί αδελφό του.
- και απαντήστε στις παρακάτω ερωτήσεις
- (α) Δώστε τα σωστά B+-δέντρα που προκύπτουν για κάθε μια από τις επιλογές.
 - (β) Συγκρίνετε τις διαφορετικές επιλογές από άποψη κόστους.
 - (γ) Υπάρχουν περιπτώσεις που θα προτιμούσατε τη μια από την άλλη επιλογή και γιατί. (Υπόδειξη: σκεφτείτε διαφορετικά ποσοστά εισαγωγών και διαγραφών, πχ. τι γίνεται αν έχουμε μόνο διαγραφές τιμών και τι αν έχουμε ίδιο ποσοστό διαγραφών και εισαγωγών).

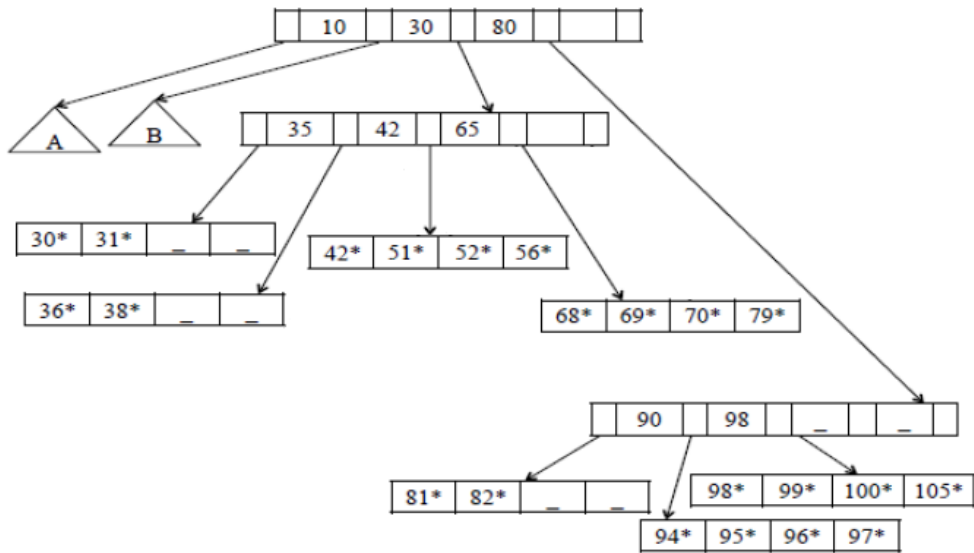


Εικόνα 1. B+ δέντρο για τις Ασκήσεις 1 και 2. Ο μέγιστος αριθμός δεικτών για τους εσωτερικούς κόμβους είναι 5 και για τα φύλλα 4. Τα φύλλα είναι διπλά διασυνδεδεμένα μεταξύ τους (δε φαίνεται στην εικόνα). Τα A, B είναι υπο-δέντρα στα οποία δείχνουν οι αντίστοιχοι δείκτες.

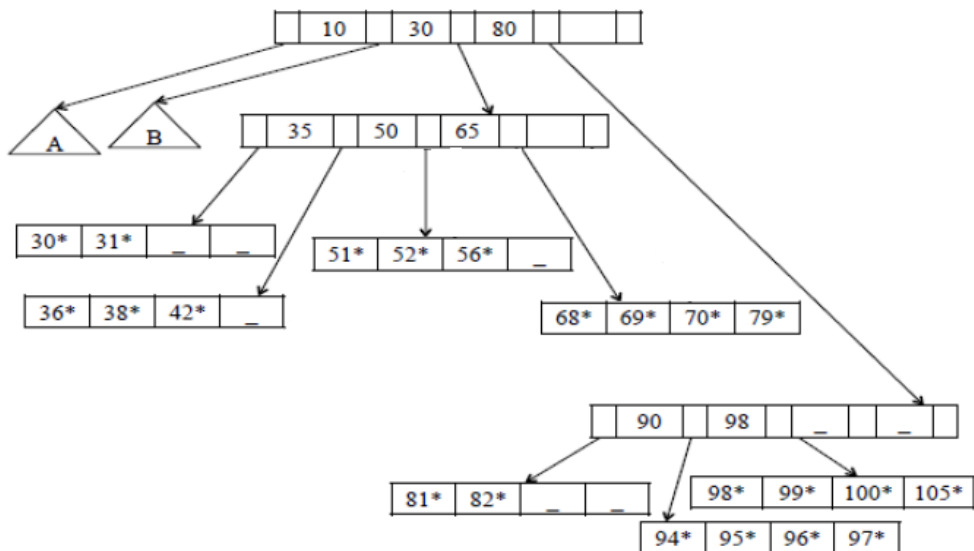
Λύση

(α)

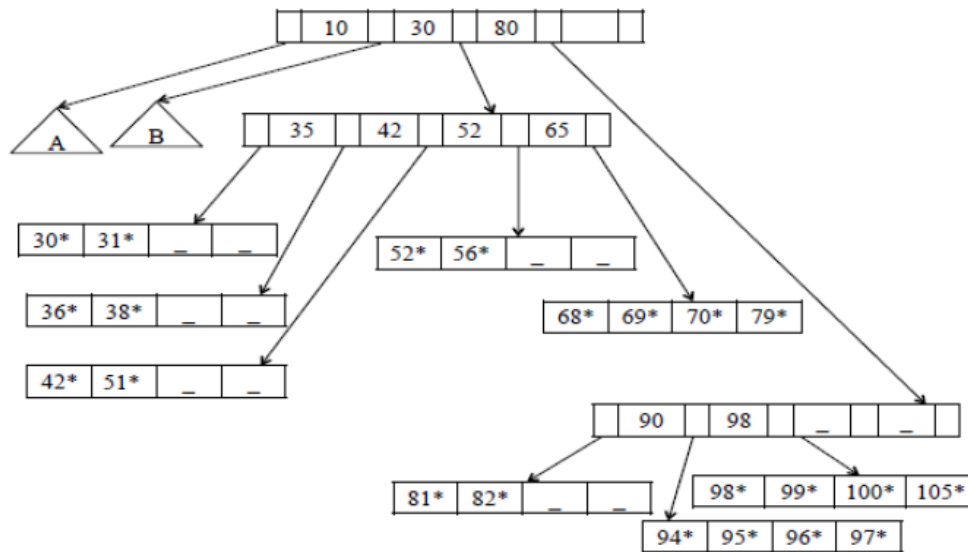
Περίπτωση 1.



Περίπτωση 2.



Περίπτωση 3.



(β)

Στην περίπτωση 1, πρώτα θα διαβαστούν οι κόμβοι από τη ρίζα ως το φύλλο που περιέχει την τιμή 43. Στη συνέχεια θα διαβαστεί ο δεξιός αδελφός του φύλλου. Έτσι θα έχουμε 4 reads. Θα ενημερωθεί ο κόμβος-αδελφός, καθώς και ο κοινός γονέας τους, και θα έχουμε 2 writes. Συνολικά θα γίνουν **6 I/Os**.

Αντίστοιχα και στην περίπτωση 2 με τον ίδιο τρόπο θα γίνουν **6 I/Os**.

Στην περίπτωση 3, θα διαβαστούν οι κόμβοι από τη ρίζα ως το φύλλο που περιέχει την τιμή 43, και ο δεξιός αδελφός του (4 reads) και θα ξαναγραφούν αυτοί οι δύο κόμβοι-φύλλα κάνοντας ανακατανομή των τιμών τους (2 writes). Τέλος, θα ενημερωθεί και ο κόμβος γονέας τους, ώστε τα κλειδιά του να ικανοποιούν τις συνθήκες που πρέπει να ισχύουν στα B+ δένδρα, δηλαδή η μεσαία τιμή κλειδιού θα διορθωθεί στον γονέα (άλλο 1 write). Συνολικά θα γίνουν **7 I/Os**.

(γ)

Αν περιμένουμε ότι θα γίνουν περισσότερες εισαγωγές στο δένδρο από ότι διαγραφές, θα θέλουμε τα φύλλα να έχουν όσο το δυνατόν περισσότερες κενές θέσεις, ώστε να μην χρειάζεται να γίνονται πολύ συχνά διασπάσεις κόμβων. Έτσι, σε αυτό το σενάριο θα προτιμούσαμε την περίπτωση 3.

Αν αντίθετα περιμένουμε ότι θα γίνουν περισσότερες διαγραφές, θα θέλουμε να μένουν όσο το δυνατόν πιο γεμάτα τα φύλλα. Σε αυτή την περίπτωση θα προτιμήσουμε τις περιπτώσεις 1 ή 2 επειδή έχουν λιγότερο κόστος ανά διαγραφή και περιμένουμε πολλές διαγραφές, και επειδή αφήνουν πιο γεμάτα τα φύλλα. Μεταξύ των δύο θα προτιμήσουμε την περίπτωση 1, διότι αφήνει πιο ισορροπημένη την κατανομή των τιμών στα φύλλα.

Η περίπτωση 3 έχει ένα I/O περισσότερο από τις δύο πρώτες, όμως οι περιπτώσεις 1 και 2 μειώνουν το πλήθος των κλειδιών του γονέα. Αυτό θα προκαλέσει συγχώνευση του γονέα-κόμβου όταν μείνει μισοάδειος, γεγονός που θα επηρεάσει και τον κόμβο γονέα του. Οι συγχωνεύσεις μπορεί να προωθηθούν αναδρομικά προς τα επάνω ως τη ρίζα, μειώνοντας έτσι το ύψος του δένδρου. Αντίστοιχα μετά από κάποιες εισαγωγές μπορεί να χρειαστεί να γίνει διάσπαση των φύλλων και των κόμβων και να αυξηθεί πάλι το ύψος. Συνεπώς, αν περιμένουμε παρόμοια ποσοστά εισαγωγών και διαγραφών προτιμούμε την περίπτωση 3, για να μην αυξομειώνουμε πολύ συχνά το δένδρο προκαλώντας πολλές λειτουργίες I/O κάθε φορά.

Άσκηση 2 [15 μονάδες]

Θεωρείστε το B+ δέντρο της Εικόνας 1 και υποθέστε ότι τα φύλλα των υπό-δέντρων A και B είναι όσο το δυνατόν πιο άδεια. Υποθέστε επίσης ότι το δέντρο ευρετηριοποιεί μια σχέση R, το πεδίο ευρετηριοποίησης είναι υπονήφιο κλειδί και κλειδί διάταξης για την R και ότι κάθε block της R χωράει 20 εγγραφές της. Δώστε μια εκτίμηση για το πόσες πλειάδες έχει η R.

Λύση

Θεωρούμε ότι τα υποδένδρα A και B έχουν ίδιο ύψος με τα άλλα υποδένδρα που φαίνονται στο σχήμα, επομένως έχουν από **3 έως 5 φύλλα** το καθένα.

Κάθε τέτοιο φύλλο είναι όσο το δυνατόν πιο άδειο, άρα έχει ακριβώς 2 δείκτες προς εγγραφές της σχέσης.

Η R είναι ταξινομημένη ως προς το υπονήφιο κλειδί, επομένως αρκεί κάθε δείκτης του ευρετηρίου να δείχνει στην 1^η εγγραφή κάθε block. Κάθε δείκτης από τα φύλλα αντιστοιχεί σε 1 block της σχέσης.

Λαμβάνουμε υπ' όψιν μας το σχήμα του δένδρου και το πλήθος κλειδιών που περιέχει και όχι τις ακριβείς τιμές των κλειδιών όπως φαίνονται στο σχήμα.

Στα 8 φύλλα όπως φαίνονται στο σχήμα περιέχονται 23 δείκτες προς blocks της σχέσης.

Στο υποδένδρο A θα υπάρχουν από $2 \cdot 3 = 6$ έως $2 \cdot 5 = 10$ δείκτες προς τη σχέση. Όμοια και για το B, άλλοι 6 έως 10 δείκτες.

Συνολικά από τα φύλλα του ευρετηρίου υπάρχουν από $23 + 6 + 6 = 35$ έως $23 + 10 + 10 = 43$ δείκτες προς τη σχέση.

Επομένως η R έχει από 35 έως 43 blocks, δηλαδή από **700 έως 860 εγγραφές**.

Άσκηση 3 [40 μονάδες]

Θεωρείστε μια σχέση r με σχήμα R(A, B, C, D, E, F) η οποία έχει 6.000.000 πλειάδες (εγγραφές) και κάθε block (σελίδα) της χωράει 10 εγγραφές. Η σχέση είναι αποθηκευμένη σε ένα αρχείο διατεταγμένο σε αύξουσα διάταξη με πεδίο διάταξης το γνώρισμα A. Το γνώρισμα A είναι υπονήφιο κλειδί για τη σχέση και παίρνει τιμές από το 0 έως 5.999.999. Για κάθε μια από τις παρακάτω τέσσερις ερωτήσεις σε σχεσιακή άλγεβρα

Ερώτηση

- (1) $\sigma_{A < 60.000}$ (r)
- (2) $\sigma_{A = 60.000}$ (r)
- (3) $\sigma_{A > 60.000 \text{ AND } A < 60.010}$ (r)
- (4) $\sigma_{A \triangleleft 60.000}$ (r)

δώστε το κόστος (σε αριθμό προσπελάσεων σελίδων) για τις παρακάτω τρεις διαφορετικές μεθόδους υπολογισμού τους και πείτε ποια μέθοδος είναι η καλύτερη σε κάθε περίπτωση.

Μέθοδος υπολογισμού

- (i) Προσπέλαση του αρχείου δεδομένων απευθείας (χωρίς χρήση ευρετηρίου).
- (ii) Χρήση ενός B+ δέντρου στο γνώρισμα A. Υποθέστε ότι κάθε εσωτερικός κόμβος έχει μέγιστο αριθμό δεικτών 100 και κάθε φύλλο έχει μέγιστο αριθμό κλειδιών 100.

(iii) Χρήση ευρετηρίου κατακερματισμού στο γνώρισμα A. Θεωρείστε μη δυναμικό κατακερματισμό (δεν υπάρχουν υπερχειλίσεις).

Λύση

Η σχέση στο δίσκο είναι αποθηκευμένη σε $(6.000.000 / 10) = 600.000$ blocks.

(i) Χωρίς Ευρετήριο

(1) Θα πρέπει να διαβαστούν όλα τα blocks της σχέσης που περιέχουν τις πρώτες 60.000 εγγραφές, με τιμές $A=0$ έως 59.999. Δηλαδή τα πρώτα $(60.000/10) = 6.000$ blocks. Το κόστος είναι **6.000** I/Os.

(2) Θα πρέπει να διαβαστούν με τη σειρά όλα τα blocks της σχέσης μέχρι το block που περιέχει την $60.001^{\text{η}}$ εγγραφή με τιμή $A=60.000$. Δηλαδή τα πρώτα $(60.001/10) = 6.001$ blocks. Το κόστος είναι **6.001** I/Os.

(3) Θα πρέπει να διαβαστούν με τη σειρά όλα τα blocks της σχέσης που περιέχουν τις πρώτες 60.010 εγγραφές (η $60.011^{\text{η}}$ έχει τιμή $A=60.010$). Δηλαδή τα πρώτα $(60.010/10) = 6.001$ blocks. Το κόστος είναι **6.001** I/Os.

(4) Θα πρέπει να διαβαστούν όλα τα blocks της σχέσης, επομένως το κόστος είναι **600.000** I/Os.

(ii) Με B+ δένδρο

Κάθε φύλλο δείχνει σε (το πολύ) 100 blocks της σχέσης. Επομένως το δένδρο χρειάζεται τουλάχιστον $[(600.000 \text{ blocks}) / (100 \text{ blocks/φύλλο})] = \mathbf{6.000}$ φύλλα.

Το επίπεδο κόμβων πάνω από τα φύλλα πρέπει να είναι τουλάχιστον $(6.000 / 100) = \mathbf{60}$ ενδιάμεσοι κόμβοι.

Στο επάνω επίπεδο αρκεί **1** κόμβος ρίζα να δείχνει τους 60 ενδιάμεσους κόμβους. Επομένως το δένδρο έχει 3 επίπεδα.

(1) Θα διαβαστούν τα blocks που ανήκουν στο μονοπάτι από τη ρίζα του δένδρου ως το 1° φύλλο που δείχνει στο 1° block της σχέσης, και ακολούθως θα διαβαστούν τα πρώτα 6.000 blocks της σχέσης.

Συνολικά θα γίνουν $3+6.000 = \mathbf{6.003}$ I/Os.

(2) Θα διαβαστούν τα blocks που ανήκουν στο μονοπάτι από τη ρίζα του δένδρου ως το φύλλο που δείχνει στο ζητούμενο block της σχέσης, και το block αυτό. Συνολικά θα γίνουν $3+1 = \mathbf{4}$ I/Os.

(3) Η εγγραφή με τιμή $A=60.001$ είναι η $60.002^{\text{η}}$ στη σχέση και βρίσκεται στο $(60.002/10) = 6.001^{\circ}$ block. Η εγγραφή με $A=60.009$ είναι η $60.010^{\text{η}}$ και βρίσκεται επίσης στο $(60.010/10) = 6.001^{\circ}$ block της σχέσης.

Θα διαβαστούν τα 3 blocks που ανήκουν στο μονοπάτι από τη ρίζα του δένδρου ως το φύλλο που δείχνει στο block της σχέσης που περιέχει την $60.002^{\text{η}}$ εγγραφή και στη συνέχεια θα διαβαστεί το block αυτό, που περιέχει της ζητούμενες εγγραφές.

Συνολικά θα γίνουν $3+1 = \mathbf{4}$ I/Os.

(4) Θα διαβαστούν τα blocks που ανήκουν στο μονοπάτι από τη ρίζα του δένδρου ως το 1^ο φύλλο που δείχνει στο 1^ο block της σχέσης, και ακολούθως θα διαβαστούν όλα τα 600.000 blocks της σχέσης. Συνολικά θα γίνουν $3+600.000 = \mathbf{600.003}$ I/Os.

(iii) Με ευρετήριο Hash

Από το προηγούμενο ερώτημα έχουμε ότι κάθε block χωράει 100 κλειδιά (και 100 δείκτες). Θεωρούμε ότι οι τιμές του κλειδιού κατανομούνται ομοιόμορφα σε $(6.000.000 / 100) = 60.000$ κάδους, ώστε να μην έχουμε υπερχείλιση με 1 block/κάδο.

Αν η συνάρτηση κατακερματισμού είναι λ.χ. η $h(x) = x \bmod 60.000$, τότε:

(1) range query: Θα πρέπει να διαβαστούν όλα τα 60.000 blocks ευρετηρίου + 6.000 blocks αρχείου που ικανοποιούν τη συνθήκη. Συνολικά **66.000** I/Os.

(2) exact query: Εφόσον δεν υπάρχει υπερχείλιση, θα βρεθεί η εγγραφή μέσω δείκτη από το block του αντίστοιχου κάδου με μόνο **2** I/Os (1 block ευρετηρίου + 1 block αρχείου).

(3) range query: Θα πρέπει να διαβαστούν όλα τα blocks ευρετηρίου, άρα 60.000 blocks ευρετηρίου + 1 block αρχείου. Συνολικά **60.001** I/Os.

(4) Θα πρέπει να προσπελαστούν όλα τα buckets. 60.000 blocks ευρετηρίου + 600.000 blocks αρχείου. Συνολικά **660.000** I/Os.

Αν η συνάρτηση κατακερματισμού είναι λ.χ. η $h(x) = x \operatorname{div} 60.000$, τότε:

(1) range query: Θα πρέπει να διαβαστούν τα blocks των πρώτων $(60.000/100) = 600$ buckets του ευρετηρίου + 6.000 blocks αρχείου που ικανοποιούν τη συνθήκη. Συνολικά **6.600** I/Os.

(2) exact query: Εφόσον δεν υπάρχει υπερχείλιση, θα βρεθεί η εγγραφή μέσω δείκτη από το block του αντίστοιχου κάδου με μόνο **2** I/Os (1 block ευρετηρίου + 1 block αρχείου).

(3) range query: Θα πρέπει να διαβαστούν τα blocks ευρετηρίου από το $(60.000/100) = 600^{\circ}$ bucket ως το $(60.010/100) = 601^{\circ}$ bucket, άρα 2 blocks ευρετηρίου + 1 block αρχείου. Συνολικά **3** I/Os.

(4) Θα πρέπει να προσπελαστούν όλα τα buckets. 60.000 blocks ευρετηρίου + 600.000 blocks αρχείου. Συνολικά **660.000** I/Os.

(iii) Με ευρετήριο Hash [b' τρόπος]

Από το προηγούμενο ερώτημα έχουμε ότι κάθε block χωράει 100 κλειδιά (και 100 δείκτες). Θεωρούμε ότι το ευρετήριο έχει δείκτες προς την 1^η εγγραφή κάθε block και οι αντίστοιχες τιμές του κλειδιού κατανομούνται ομοιόμορφα σε $(600.000 / 100) = 6.000$ κάδους, ώστε να μην έχουμε υπερχείλιση με 1 block/κάδο.

Αν η συνάρτηση κατακερματισμού είναι λ.χ. η $h(x) = x \bmod 6.000$, τότε:

(1) range query: Θα πρέπει να διαβαστούν όλα τα 6.000 blocks ευρετηρίου + 6.000 blocks αρχείου που ικανοποιούν τη συνθήκη. Συνολικά **12.000** I/Os.

(2) exact query: Εφόσον δεν υπάρχει υπερχειλίση, θα βρεθεί η εγγραφή μέσω δείκτη από το block του αντίστοιχου κάδου με μόνο **2** I/Os (1 block ευρετηρίου + 1 block αρχείου).

(3) range query: Θα πρέπει να διαβαστούν όλα τα blocks ευρετηρίου, άρα 6.000 blocks ευρετηρίου + 1 block αρχείου. Συνολικά **6.001** I/Os.

(4) Θα πρέπει να προσπελαστούν όλα τα buckets. 6.000 blocks ευρετηρίου + 600.000 blocks αρχείου. Συνολικά **606.000** I/Os.

Αν η συνάρτηση κατακερματισμού είναι λ.χ. η $h(x) = x \div 6.000$, τότε:

(1) Όπως υπολογίστηκε στο ερώτημα (iii), η 60.000^{th} εγγραφή βρίσκεται στο 6.000° block του αρχείου. Ο δείκτης που δείχνει σε αυτό το block του αρχείου βρίσκεται στο $(6.000/100) = 60^{\circ}$ bucket του ευρετηρίου. Θα πρέπει να διαβαστούν τα blocks των πρώτων 60 buckets ευρετηρίου + 6.000 blocks αρχείου που ικανοποιούν τη συνθήκη. Συνολικά **6.060** I/Os.

(2) exact query: Εφόσον δεν υπάρχει υπερχειλίση, θα βρεθεί η εγγραφή μέσω δείκτη από το block του αντίστοιχου κάδου με μόνο **2** I/Os (1 block ευρετηρίου + 1 block αρχείου).

(3) Όπως υπολογίστηκε στο ερώτημα (iii), οι εγγραφές από την 60.001^{th} ως την 60.010^{th} εγγραφή είναι αποθηκευμένες στο $(60.001/10) = 6.001^{\circ}$ block του αρχείου. Ο δείκτης ευρετηρίου που δείχνει στο 6.001° block του αρχείου βρίσκεται στο $(6.001/100) = 61^{\circ}$ bucket ευρετηρίου. Συνολικά θα γίνουν **2** I/Os (1 block ευρετηρίου + 1 block αρχείου).

(4) Θα πρέπει να προσπελαστούν όλα τα buckets. 6.000 blocks ευρετηρίου + 600.000 blocks αρχείου. Συνολικά **606.000** I/Os.

Άσκηση 4 [20 μονάδες]

Έστω ότι έχετε μια σχέση με 100.000 εγγραφές. Επιθυμείτε να κατακερματίσετε τη σχέση σε ένα πίνακα κατακερματισμού με 1.000 κάδους. Ένα μπλοκ δίσκου μπορεί να αποθηκεύσει 100 εγγραφές (μαζί με ένα επιπρόσθετο δείκτη προς ένα μπλοκ υπερχειλίσης). Θεωρείστε ότι ένα μπλοκ δίσκου δεν μπορεί να αποθηκεύσει εγγραφές από δύο διαφορετικούς κάδους.

- (α) Ποιος είναι ο μέγιστος αριθμός από μπλοκ δίσκου που απαιτούνται για την αποθήκευση της σχέσης;
- (β) Ποιος είναι ο ελάχιστος αριθμός από μπλοκ δίσκου που απαιτούνται για τη σχέση;
- (γ) Ποια είναι η απάντηση στο ερώτημα (α) αν η σχέση είχε 100.099 εγγραφές;

Λύση

(α)

Στην χειρότερη περίπτωση αποθηκεύεται 1 εγγραφή/κάδο στους 999 κάδους, και όλες οι υπόλοιπες $100.000 - 999 = 99.001$ εγγραφές αποθηκεύονται όλες σε ένα κάδο.

Έτσι για τους 999 κάδους δεσμεύεται από 1 block στον καθένα, που θα περιέχει 1 εγγραφή. Συνολικά 999 blocks.

Στον τελευταίο κάδο θα έχουμε *υπερχείλιση*. Επειδή δεν επιτρέπεται κάποιο block δίσκου να περιέχει εγγραφές από διαφορετικούς κάδους, δεν μπορούμε να έχουμε ανοιχτή διεθυνσιοδότηση, ούτε πολλαπλό κατακερματισμό. Έτσι θα έχουμε αλυσιδωτή σύνδεση με blocks υπερχείλισης στον κάδο αυτό.

Για να αποθηκευθούν οι υπόλοιπες εγγραφές θα χρειαστούν $(99.001 / 100) = 991$ blocks. (990 γεμάτα και 1 με μια εγγραφή).

Συνολικά για την αποθήκευση της σχέσης θα χρειαστούν το πολύ $999 + 991 =$ **1.990 blocks**.

(β)

Στην καλύτερη περίπτωση όλες οι εγγραφές μπορούν να κατανεμηθούν ομοιόμορφα στους 1.000 κάδους, θα έχουμε $100.000 / 1.000 = 100$ εγγραφές/κάδο.

Άρα, αρκεί 1 block ανά κάδο, που θα είναι γεμάτο με 100 εγγραφές.

Συνολικά αρκούν **1.000 blocks** (όσα και οι κάδοι).

(γ)

Ομοίως θα αποθηκευθεί 1 εγγραφή/κάδο σε 999 κάδους, και όλες οι υπόλοιπες $100.099 - 999 = 99.100$ εγγραφές αντιστοιχούν όλες σε ένα κάδο.

Για τους 999 κάδους δεσμεύεται από 1 block στον καθένα, που θα περιέχει 1 εγγραφή. Συνολικά 999 blocks.

Στον τελευταίο κάδο θα έχουμε *υπερχείλιση* και θα κάνουμε αλυσιδωτή σύνδεση με blocks υπερχείλισης. Για να αποθηκευθούν οι υπόλοιπες εγγραφές θα χρειαστούν $(99.100 / 100) = 991$ blocks. (όλα γεμάτα με εγγραφές).

Συνολικά για την αποθήκευση της σχέσης θα χρειαστούν και σε αυτή την περίπτωση το πολύ **1.990 blocks**.