ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΛΥΣΗ ΤΗΣ ΔΕΥΤΕΡΗΣ ΑΣΚΗΣΗΣ

ΜΑΘΗΜΑ **ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**
ΑΚΑΔ. ΕΤΟΣ 2010-11

ΔΙΔΑΣΚΟΝΤΕΣ Ιωάννης Βασιλείου *Καθηγητής*, Τομέας Πληροφορικής
Τιμολέων Σελλής *Καθηγητής*, Τομέας Πληροφορικής

Ερώτημα 1.

Θεωρείστε ένα αρχικά άδειο B^+– δένδρο όπου κάθε ενδιάμεσος κόμβος μπορεί να περιέχει μέχρι 4 δείκτες.

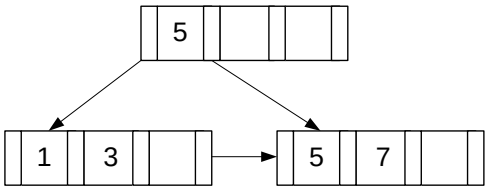
(α) Δώστε τη μορφή του δένδρου που προκύπτει σε κάθε βήμα κατά την εισαγωγή των παρακάτω κλειδιών με την ακόλουθη σειρά: 1, 5, 7, 3, 8, 2, 9, 4, 10, 6, 11. Θεωρείστε ότι η εισαγωγή γίνεται βήμα-βήμα χωρίς bulkloading. Ποιος είναι ο βαθμός πληρότητας (%) του δένδρου που προκύπτει; Για τον υπολογισμό του βαθμού πληρότητας διαιρέστε τον αριθμό των κλειδιών σε όλα τα επίπεδα του δένδρου με τον μέγιστο αριθμό κλειδιών που θα μπορούσαν να χωρέσουν στο ένα δένδρο με το ίδιο ύψος και τάξη.
(β) Δώστε τη μορφή του δένδρου που προκύπτει σε κάθε βήμα κατά τη διαγραφή των κλειδιών με την ακόλουθη σειρά: 5, 2, 10, 4, 1, 3, 8, 9, 6, 7, 11.

ΛΥΣΗ

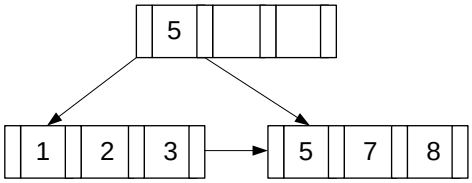
(α) Σύμφωνα με τον ορισμό του B+-δένδρου κάθε κόμβος εκτός της ρίζας μπορεί να έχει από 2 έως 4 δείκτες, ενώ κάθε φύλλο θα έχει από 2 έως 3 κλειδιά.

Αρχικά εισάγονται τα κλειδιά 1, 5, 7 τα οποία καταχωρούνται στη ρίζα.

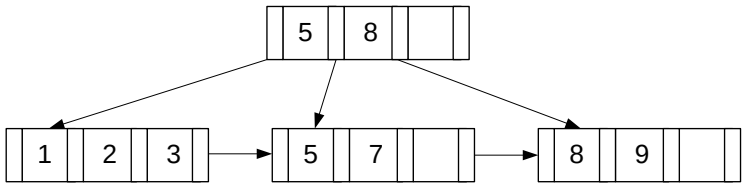
Η εισαγωγή του 3 προκαλεί τη διάσπαση της ρίζας σε δύο κόμβους, και το κλειδί 5 ανεβαίνει ως κλειδί της νέας ρίζας.



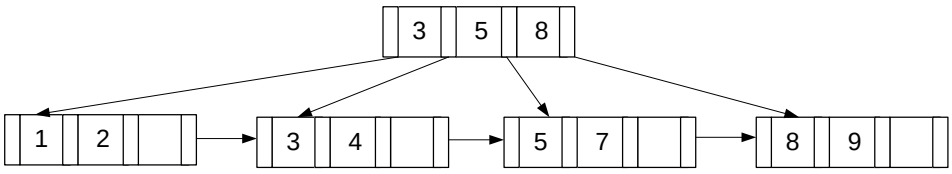
Τα κλειδιά 8 και 2 εισάγονται κανονικά στο δεξιό και αριστερό φύλλο.



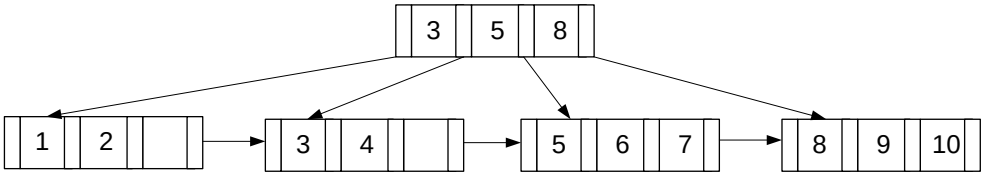
Η εισαγωγή του 9 προκαλεί τη διάσπαση του τελευταίου φύλλου και το 8 ανεβαίνει στη ρίζα.



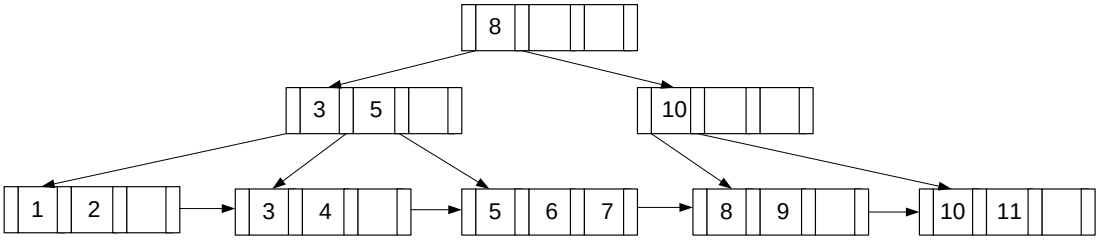
Η εισαγωγή του 4 προκαλεί τη διάσπαση του αριστερού φύλλου και το 3 ανεβαίνει στη ρίζα.



Η εισαγωγή των 10 και 6 γίνεται κανονικά στα φύλλα.

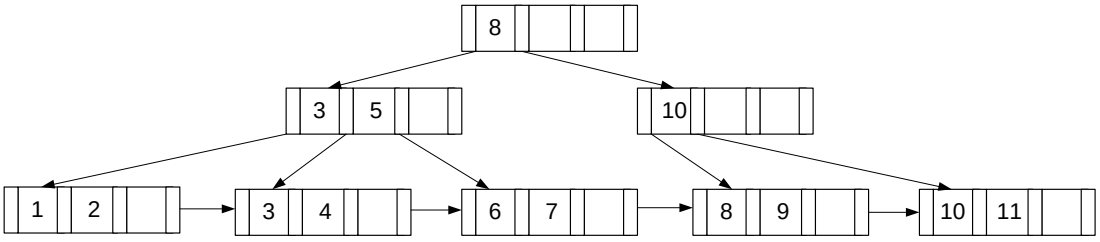


Η εισαγωγή του 11 προκαλεί διάσπαση του τελευταίου φύλλου. Το κλειδί 10 θα πρέπει να ανέβει στη ρίζα. Επειδή η ρίζα είναι ήδη γεμάτη προκαλείται νέα διάσπαση και το 8 ανεβαίνει και γίνεται κλειδί της νέας ρίζας.



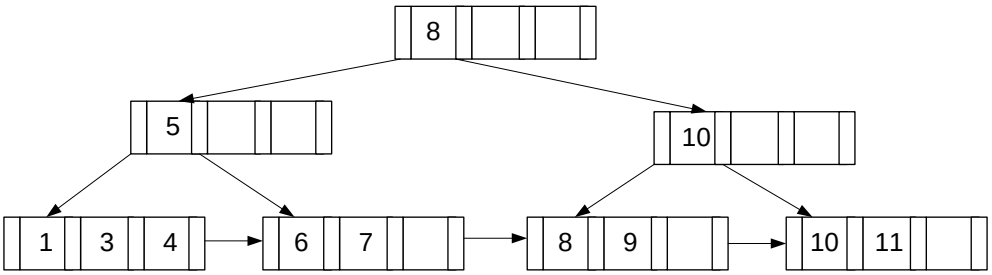
Το B+-δένδρο περιλαμβάνει 15 κλειδιά. Συνολικά, με ένα δένδρο ίδιου ύψους και τάξης, ο μέγιστος αριθμός κλειδιών προκύπτει ίσος με 3 (ρίζα) + 4*3 ($1^{ο}$ επίπεδο) + 4*4*3 (φύλλα) = 63 κλειδιά. Συνεπώς, ο βαθμός πληρότητας του δένδρου είναι 15/63 x100%= **23.8%**

(β) Η διαγραφή του 5 γίνεται χωρίς να επιφέρει καμία αλλαγή στη δομή του B+-δένδρου.

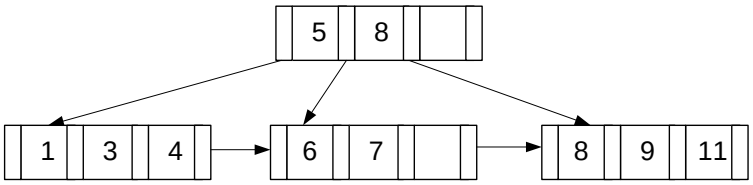


Η διαγραφή του 2 αφήνει το αριστερό φύλλο μόνο με ένα κλειδί. Αυτό αντίκειται στη συνθήκη ότι κάθε φύλλο πρέπει να έχει από 1 έως 3 κλειδιά. Συνεπώς το αριστερό φύλλο

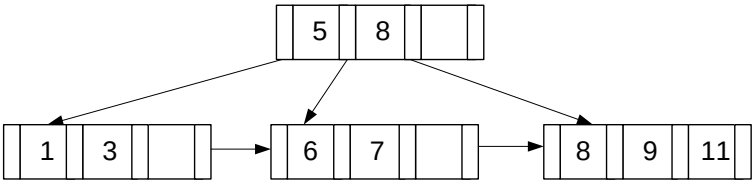
πρέπει να συγχωνευτεί με το διπλανό. Επιπλέον το κλειδί 3 φεύγει και από το πιο πάνω επίπεδο κόμβων, ο οποίος μένει με το κλειδί 5.



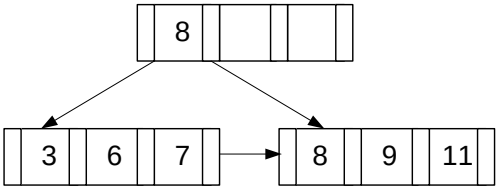
Η διαγραφή του 10 αφήνει το τελευταίο φύλλο μόνο με ένα κλειδί, επομένως πρέπει να συγχωνευτεί με το διπλανό φύλλο. Επιπλέον, θα πρέπει να διαγράψουμε και τον δείκτη προς τον κόμβο που διαγράφηκε από το πιο πάνω επίπεδο. Η διαγραφή του δείκτη αφήνει τον κόμβο με κλειδί 10 μόνο με έναν δείκτη. Συνεπώς πρέπει να συγχωνευτεί επίσης με τον διπλανό του κόμβο. Η συγχώνευση αυτή έχει επίσης ως συνέπεια να διαγραφεί ο δείκτης προς το δεξί υπόδενδρο από τη ρίζα. Αυτή η διαγραφή έχει ως συνέπεια να μείνει η ρίζα μόνο με έναν δείκτη επομένως πρέπει να μειωθεί το ύψος του δένδρου κατά ένα επίπεδο.



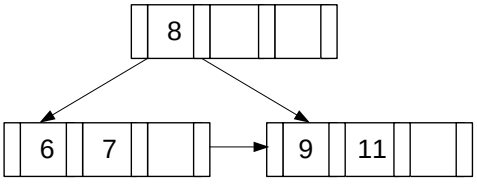
Η διαγραφή του 4 δεν επιφέρει καμία αλλαγή στη δομή του B+-δένδρου.



Η διαγραφή του 1 αφήνει το αριστερό φύλλο μόνο με ένα κλειδί επομένως πρέπει να συγχωνευτεί με το διπλανό φύλλο. Επιπλέον, ο δείκτης προς το φύλλο και το κλειδί 5 πρέπει επίσης να διαγραφούν από τη ρίζα.



Οι διαγραφές των 3 και 8 δεν επιφέρουν καμία αλλαγή στη δομή του δένδρου.



Η διαγραφή του 9 αφήνει το δεξί φύλλο μόνο με 1 κλειδί και επομένως τα δύο φύλλα πρέπει να συγχωνευτούν. Ως αποτέλεσμα ο ένας δείκτης πρέπει να διαγραφεί από τη ρίζα και η ρίζα μένει μόνο με έναν δείκτη. Συνεπώς το ύψος του B+-δένδρου μειώνεται κατά ένα επίπεδο και το φύλλο γίνεται ρίζα.

Οι διαγραφές των υπολοίπων κλειδιών είναι απλές και δεν επιφέρουν καμία αλλαγή, μέχρι το B+-δένδρο να αδειάσει εντελώς.

Ερώτημα 2.

Θεωρήστε μία σχέση R(A, B, C) με 50000 εγγραφές η οποία είναι ταξινομημένη με κλειδί το γνώρισμα A που παίρνει τιμές από 0 έως 49999. Υποθέστε ότι θέλουμε να δεικτοδοτήσουμε τη σχέση με κλειδί το A χρησιμοποιώντας ένα B^+– δένδρο. Τα φύλλα του δένδρου αποθηκεύουν δείκτες και όχι τις πραγματικές εγγραφές της σχέσης. Επίσης θεωρείστε ότι:

- κάθε δείκτης έχει μέγεθος 10 bytes,
- κάθε κλειδί έχει μέγεθος 200 bytes,
- κάθε εγγραφή έχει μέγεθος 500 bytes,
- το μέγεθος κάθε block στο δίσκο είναι 4 KB

(α) Ποιο είναι το ύψος (αριθμός επιπέδων) που έχει το ελάχιστο B^+– δένδρο που μπορεί να δεικτοδοτήσει ολόκληρη τη σχέση;

(β) Προσδιορίστε τον αριθμό των λειτουργιών I/O που απαιτούνται για να απαντηθεί ένα ερώτημα της μορφής: «Βρες όλες τις εγγραφές της R με A=25000».

(γ) Αν κάθε φύλλο είναι κατά 70% γεμάτο, ποιο είναι το ύψος του νέου B^+– δένδρου που προκύπτει;

(δ) Αν η σχέση δεν είναι ταξινομημένη ως προς το γνώρισμα A, τι ύψος έχει το νέο B^+– δένδρο που προκύπτει;

ΛΥΣΗ

(α) Αρχικά υπολογίζουμε πόσα blocks χρειάζονται για να αποθηκεύσουμε ολόκληρη τη σχέση R.

Κάθε εγγραφή έχει μέγεθος 500 bytes και το μέγεθος κάθε block είναι 4096 bytes. Επομένως σε κάθε block μπορούν να αποθηκευτούν έως floor(4096/500) = 8 εγγραφές.

Η σχέση έχει συνολικά 50000 εγγραφές και κάθε block χωράει 8 εγγραφές. Επομένως χρειάζονται συνολικά ceil(50000/8) = 6250 blocks.

Η σχέση R είναι ταξινομημένη ως προς το γνώρισμα A. Επομένως μπορούμε να έχουμε δείκτες μόνο στην πρώτη εγγραφή κάθε block. Άρα χρειαζόμαστε τόσους δείκτες στο δίσκο όσος είναι ο αριθμός των blocks που καταλαμβάνει η σχέση στο δίσκο, άρα χρειάζονται 6250 δείκτες.

Έστω ότι κάθε φύλλο περιέχει d κλειδιά, d δείκτες προς blocks του δίσκου και 1 δείκτη προς το επόμενο φύλλο. Κάθε φύλλο έχει μέγεθος όσο ένα block. Λαμβάνοντας υπόψη το μέγεθος κλειδιού (200 bytes) και το μέγεθος κάθε δείκτη (10 bytes), μπορούμε να βρούμε τον αριθμό κλειδιών που χωρούν σε ένα φύλλο λύνοντας την παρακάτω εξίσωση ως προς d:

d(200 + 10) + 10 ≤ 4096, άρα d = 19

Έχουμε σε κάθε φύλλο 19 δείκτες προς blocks. Συνεπώς (αν όλα τα φύλλα είναι γεμάτα) χρειάζονται τουλάχιστον ceil(6250/19) = 329 φύλλα για να δεικτοδοτηθεί ολόκληρη η σχέση.

Σε κάθε ενδιάμεσο κόμβο του B+-δένδρου, υπάρχουν d+1 = 20 δείκτες.

Έστω ότι το ζητούμενο ύψος του B+-δένδρου είναι h. Τότε ο αριθμός κόμβων στο επίπεδο h είναι ίσος με 20^{h-1}. Για το ελάχιστο ύψος h πρέπει να ισχύει $20^{h-1} \geq 329$, άρα **h = 3**.

(β) Για να βρούμε την εγγραφή πρέπει να διασχίσουμε ένα μονοπάτι από τη ρίζα του B+ δένδρου μέχρι το αντίστοιχο φύλλο. Το μονοπάτι έχει μήκος h, συνεπώς χρειάζονται 3 λειτουργίες I/O πάνω στο δένδρο. Επιπλέον χρειάζεται άλλη μία λειτουργία για να ανακτήσουμε το συγκεκριμένο block από τον δίσκο. Άρα συνολικά χρειάζονται **4 I/Os.**

(γ) Αν τα φύλλα είναι γεμάτα κατά 70% τότε κάθε φύλλο περιέχει 0,7*19 = 13 δείκτες.

Συνεπώς χρειάζονται τουλάχιστον ceil(6250/13) = 481 φύλλα για να δεικτοδοτηθεί ολόκληρη η σχέση.

Αντιστοίχως με το (α), εφόσον d = 13 σε κάθε ενδιάμεσο κόμβο του B+-δένδρου, υπάρχουν d+1 = 14 δείκτες.

Έστω ότι το ζητούμενο ύψος του B+-δένδρου είναι h. Τότε ο αριθμός κόμβων στο επίπεδο h είναι ίσος με 14^{h-1}. Άρα για το ελάχιστο ύψος h θα ισχύει $14^{h-1} \geq 481$, άρα **h = 4**.

(δ) Αν η σχέση που δεικτοδοτούμε δεν είναι ταξινομημένη ως προς το A, τότε χρειάζεται να δεικτοδοτήσουμε όλες τις εγγραφές της R στο δίσκο. Συνεπώς χρειάζεται να δεικτοδοτήσουμε 50000 εγγραφές και επομένως θα έχουμε τουλάχιστον ceil(50000/19) = 2632 φύλλα. Για το ελάχιστο ύψος h θα ισχύει $20^{h-1} \geq 2632$, άρα **h = 4**.

Ερώτημα 3.

(α) Θεωρήστε γραμμικό κατακερματισμό όπου κάθε κάδος χωράει έως 2 εγγραφές και ως συνάρτηση κατακερματισμού τη συνάρτηση h(x) = x mod 3. Δώστε τη μορφή του ευρετηρίου που προκύπτει σε κάθε βήμα κατά την εισαγωγή των παρακάτω κλειδιών: 1, 5, 8, 6, 9, 12, 17, 4, 14.

(β) Έστω μία σχέση R με 10000 εγγραφές και κλειδί το γνώρισμα A που παίρνει τιμές από 0 έως 9999. Θέλουμε να οργανώσουμε τη σχέση ως ένα κατακερματισμένο αρχείο με 500 κάδους. Κάθε εγγραφή έχει μέγεθος 100 bytes ενώ το μέγεθος του block είναι 4 KB.

(i) Ποιος είναι ο ελάχιστος και ποιος ο μέγιστος αριθμός blocks που απαιτούνται για την αποθήκευση της σχέσης R; Υπολογίστε τον παράγοντα φόρτωσης L σε κάθε περίπτωση. Για τον υπολογισμό του L διαιρέστε τον αριθμό των εγγραφών που αποθηκεύονται με τον μέγιστο αριθμό των εγγραφών που μπορούν να χωρέσουν σε όλους τους κάδους.

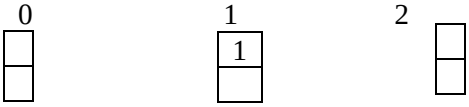
(ii) Προσδιορίστε τον ελάχιστο και τον μέγιστο αριθμό των λειτουργιών I/O που απαιτούνται για να απαντηθεί ένα ερώτημα της μορφής: «Βρες όλες τις εγγραφές της R με A=5000»

ΛΥΣΗ

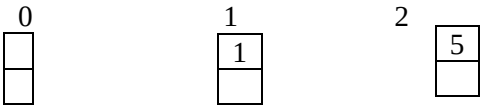
(α) Αρχικά τα hash values των κλειδιών είναι:

key	h(key)
1	1
5	2
8	2
6	0
9	0
12	0
17	2
4	1
14	2

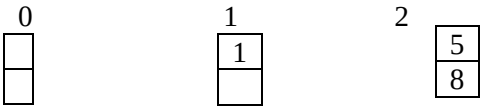
Εισαγωγή 1:



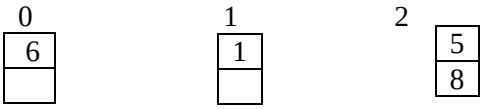
Εισαγωγή 5:



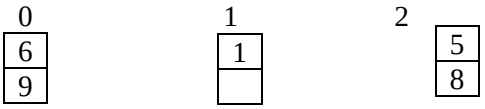
Εισαγωγή 8:



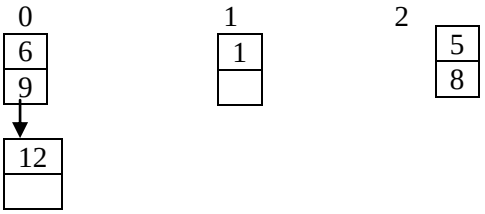
Εισαγωγή 6:



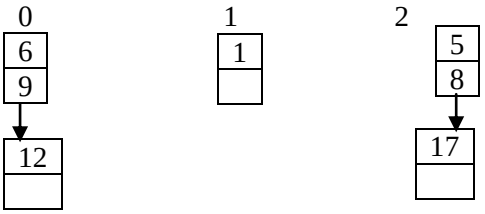
Εισαγωγή 9:



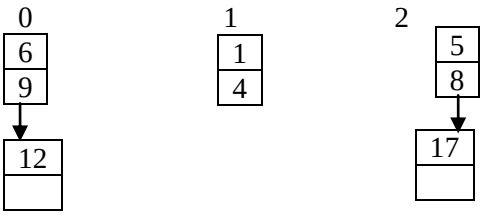
Εισαγωγή 12: Απαιτείται κάδος υπερχείλισης



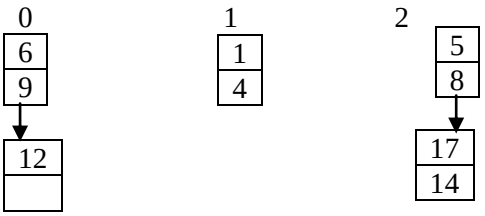
Εισαγωγή 17: Απαιτείται block υπερχείλισης



Εισαγωγή 4:



Εισαγωγή 14:



(β) (i) Το μέγεθος κάθε block είναι 4096 bytes και το μέγεθος κάθε εγγραφής 500 bytes. Συνεπώς σε κάθε block χωρούν μέχρι floor(4096/100) = 40 εγγραφές.

Στην καλύτερη περίπτωση χρειάζονται ακριβώς 10000/40 = 250 blocks για να αποθηκεύσουν το αρχείο. Συνεπώς, αν η συνάρτηση hash που έχει επιλεγεί κατανέμει τις εγγραφές ισομερώς στα μισά buckets και αφήσει άδεια τα άλλα μισά, χρειαζόμαστε συνολικά μόνο **250 blocks**. Θεωρούμε ότι τα άδεια buckets δεν δεσμεύουν blocks συνεπώς για τον υπολογισμό του L λαμβάνουμε υπόψη μόνο τα buckets με τουλάχιστον μία εγγραφή.

Ο παράγοντας φόρτωσης είναι ίσος με [10000 / (250*40)]100% = **100%**.

Στη χειρότερη περίπτωση υπάρχει 1 ακριβώς εγγραφή σε όλα τα buckets εκτός ενός. Στο bucket αυτό πρέπει να αποθηκευτούν οι υπόλοιπες 10000 – 499 = 9501 εγγραφές. Κάθε block

περιέχει 40 εγγραφές, συνεπώς χρειαζόμαστε ceil(9501/40) = 238 blocks (1 κανονικό + 237 blocks υπερχείλισης) σε αυτό το bucket. Άρα έχουμε συνολικά 500 blocks + 237 blocks υπερχείλισης = **737 blocks**.

Ο παράγοντας φόρτωσης είναι ίσος με [10000/(737*40)]100% = **34%.**

(ii) Στην καλύτερη περίπτωση, όταν η ζητούμενη η εγγραφή δεν βρίσκεται σε block υπερχείλισης, χρειάζεται μόλις **1 λειτουργία I/O** για να ανακτηθεί η εγγραφή.

Στη χειρότερη περίπτωση, όλες οι εγγραφές έχουν αποθηκευτεί σε ένα bucket. Στην περίπτωση αυτή το ένα bucket θα έχει 250 blocks και τα υπόλοιπα 499 buckets θα είναι άδεια. Αν η ζητούμενη εγγραφή βρίσκεται αποθηκευμένη στο τελευταίο block υπερχείλισης τότε θα χρειαστούν ακριβώς **250 I/Os**.