



ΕΡΓΑΣΤΗΡΙΟ #3

ΜΕΡΟΣ Α': Tree Structured Indexing

1. **(Exercise 10.1)** Consider the B+ tree index of order $d = 2$ shown in Figure 10.1.
 - i. Show the tree that would result from inserting a data entry with key 9 into this tree.
 - ii. Show the B+ tree that would result from inserting a data entry with key 3 into the original tree. How many page reads and page writes does the insertion require?
 - iii. Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the left sibling is checked for possible redistribution.
 - iv. Show the B+ tree that would result from deleting the data entry with key 8 from the original tree, assuming that the right sibling is checked for possible redistribution.
 - v. Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 46 and then deleting the data entry with key 52.
 - vi. Show the B+ tree that would result from deleting the data entry with key 91 from the original tree.
 - vii. Show the B+ tree that would result from starting with the original tree, inserting a data entry with key 59, and then deleting the data entry with key 91.
 - viii. Show the B+ tree that would result from successively deleting the data entries with keys 32, 39, 41, 45, and 73 from the original tree.

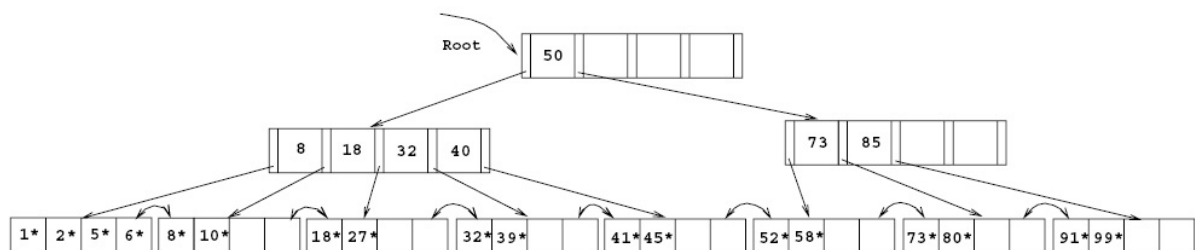


Figure 10.1 Tree for Exercise 10.1

Απαντήσεις:

- i. Η καταχώριση με κλειδί 9 παρεμβάλλεται στο δεύτερο φύλλο. Το δέντρο που προκύπτει μετά την καταχώριση παρουσιάζεται στο σχήμα.

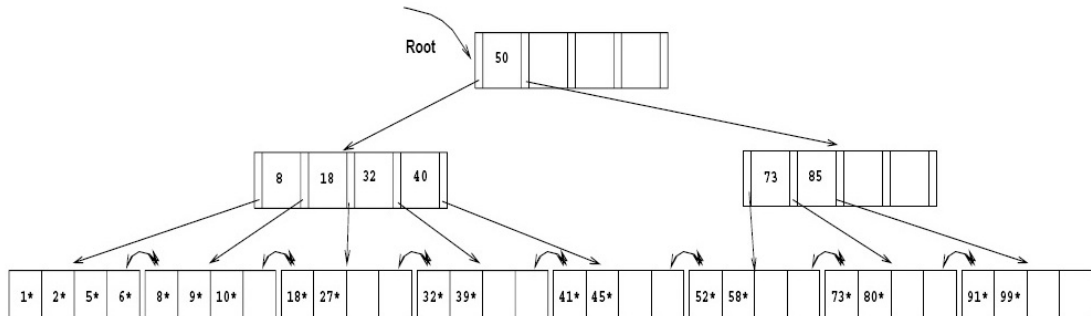


Figure 10.2

- ii. Η καταχώριση με κλειδί 3 πηγαίνει στο πρώτο φύλλο F . Δεδομένου ότι το F μπορεί να χωρέσει το πολύ τέσσερις καταχωρίσεις ($d = 2$), το F διασπάται. Η χαμηλότερη καταχώριση του νέου φύλλου ανεβαίνει στον πρόγονο του F που διασπάται επίσης. Το αποτέλεσμα φαίνεται στο σχήμα 10.3. Η εισαγωγή θα απαιτήσει να γραφούν 5 σελίδες, διάβασμα 4 σελίδων και δημιουργία 2 νέων σελίδων.

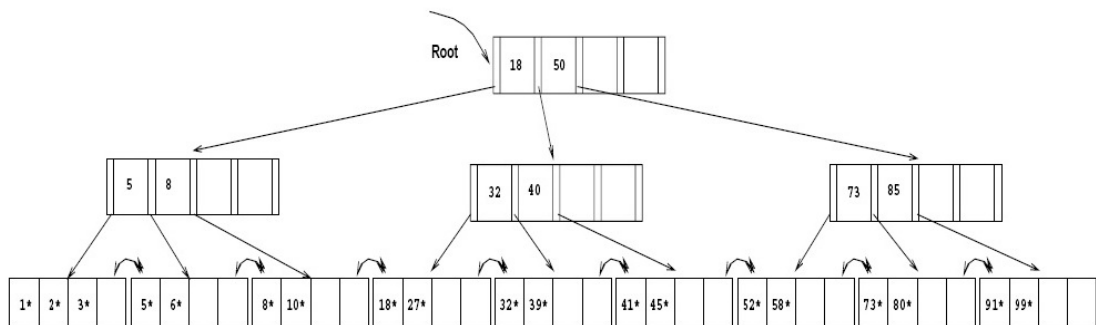


Figure 10.3

- iii. Η καταχώριση με κλειδί 8 διαγράφεται, με συνέπεια το φύλλο N να έχει λιγότερο από δύο καταχωρίσεις. Το αριστερό sibling L ελέγχεται για ανακατανομή. Δεδομένου ότι το L έχει περισσότερες από δύο καταχωρίσεις, τα υπόλοιπα κλειδιά ανακατανέμονται μεταξύ του L και του N , με συνέπεια το δέντρο στο σχήμα 10.4.

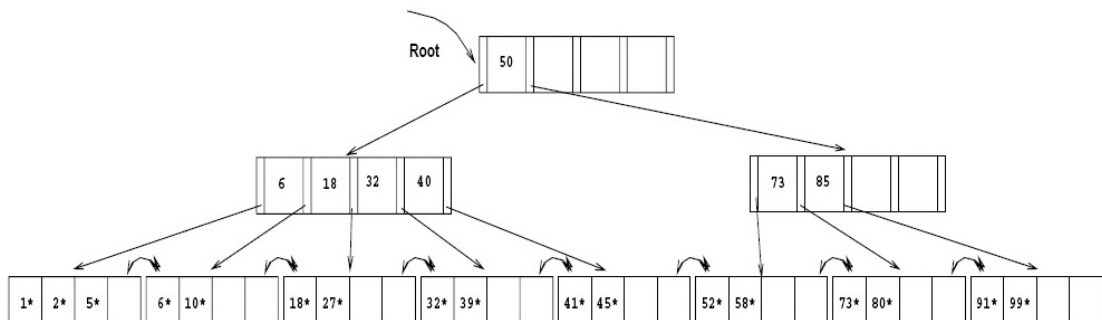


Figure 10.4

- iv. Όπως και στο ερώτημα 3, η καταχώριση με κλειδί 8 διαγράφεται από το sibling φύλλο N. Το δεξιά sibling φύλλο R του N ελέγχεται για ανακατανομή, αλλά το R έχει τον ελάχιστο αριθμό κλειδιών. Επομένως τα δυο siblings συγχωνεύονται. Το κλειδί στον πρόγονο των N και R που ξεχώριζε τα πρόσφατα συγχωνευμένα φύλλα διαγράφεται. Το προκύπτον δέντρο παρουσιάζεται στο σχήμα 10.5.

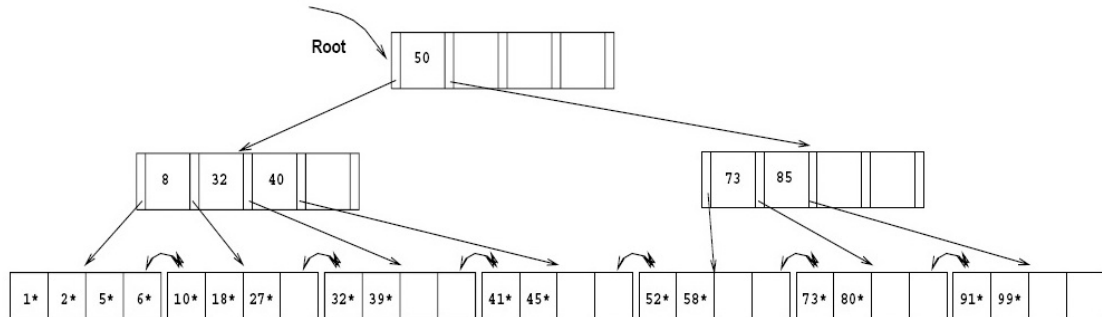


Figure 10.5

- v. Η καταχώριση με κλειδί 46 μπορεί να παρεμβληθεί χωρίς οποιεσδήποτε δομικές αλλαγές στο δέντρο. Αλλά η αφαίρεση της καταχώρισης με κλειδί 52 αναγκάζει το φύλλο L να συγχωνευτεί με ένα sibling (επιλέξαμε το δεξιά sibling). Αυτό οδηγεί στην αφαίρεση ενός κλειδιού στον πρόγονο A του L και με αυτόν τον τρόπο του χαμηλώματος του αριθμού κλειδιών στο A κάτω από τον ελάχιστο αριθμό κλειδιών. Δεδομένου ότι το αριστερό sibling B του A έχει περισσότερα από τον ελάχιστο αριθμό κλειδιών, πραγματοποιείται ανακατανομή μεταξύ του A και του B. Το τελικό δέντρο απεικονίζεται στο σχήμα 10.6.

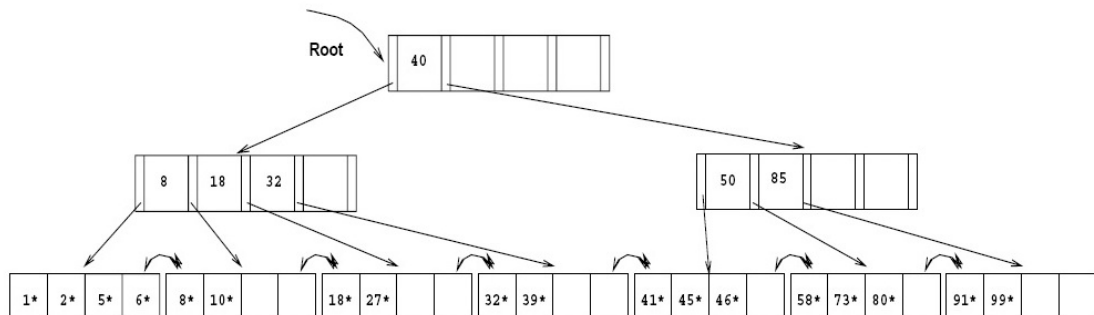


Figure 10.6

- vi. Η διαγραφή της καταχώρισης με κλειδί 91 προκαλεί ένα σενάριο παρόμοιο με το ερώτημα 5. Το αποτέλεσμα φαίνεται στο σχήμα 10.7.

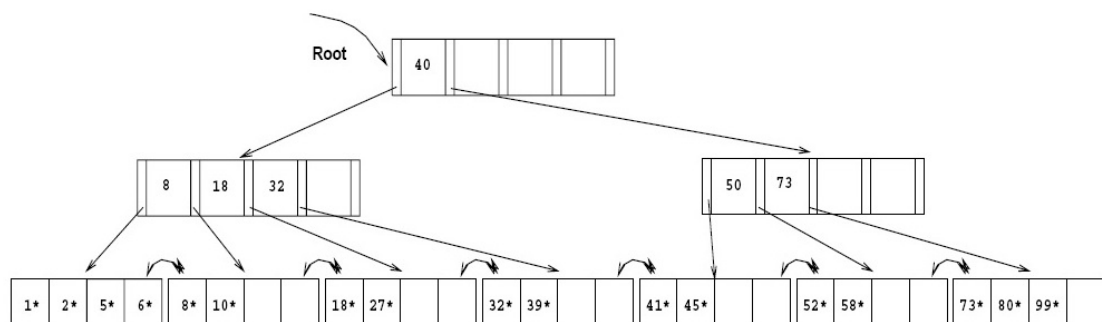


Figure 10.7

vii. Η καταχώριση με το κλειδί 59 μπορεί να εισαχθεί χωρίς οποιεσδήποτε δομικές αλλαγές στο δέντρο. Κανένα sibling του φύλλου με την καταχώριση με το κλειδί 91 δεν επηρεάζεται από την εισαγωγή. Επομένως η διαγραφή της καταχώρισης με κλειδί 91 αλλάζει το δέντρο με τρόπο πολύ παρόμοιο με το ερώτημα 6. Το αποτέλεσμα απεικονίζεται στο σχήμα 10.8.

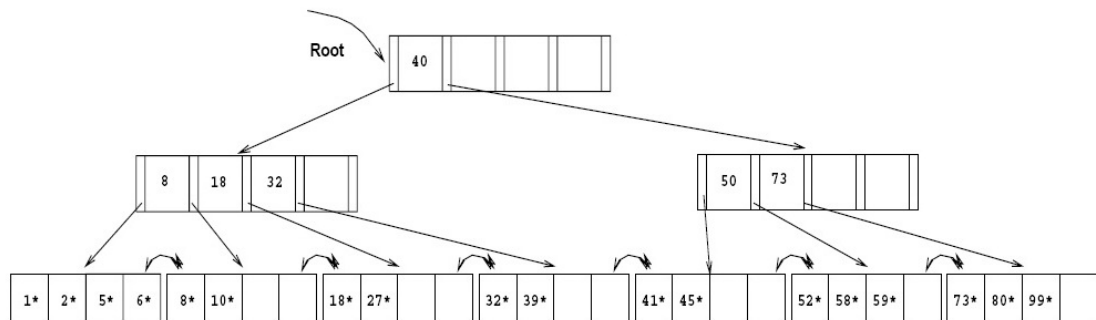


Figure 10.8

viii. Υπολογίζοντας το δεξιά sibling για πιθανή συγχώνευση πρώτα, η διαδοχική διαγραφή των καταχωρίσεων με κλειδιά 32, 39, 41, 45 και 73 παράγουν το δέντρο που παρουσιάζεται στο σχήμα 10.9.

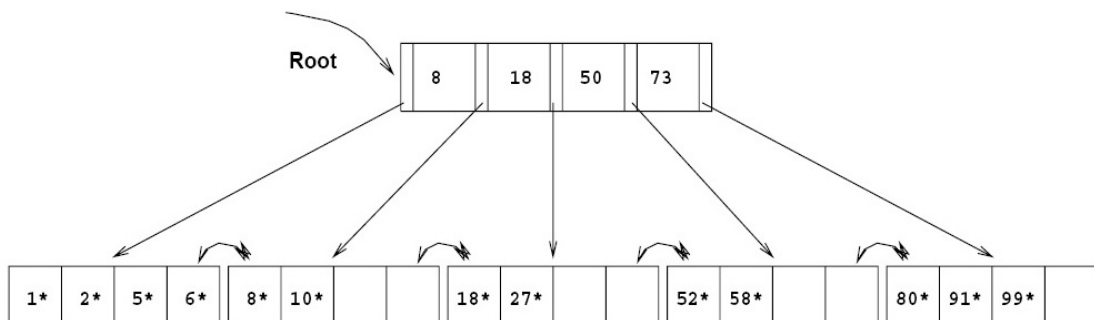


Figure 10.9

ΜΕΡΟΣ Β': Extendible Hashing

2. **(Exercise 11.1)** Consider the Extendible Hashing index shown in Figure 11.1. Answer the following questions about this index:
- What can you say about the last entry that was inserted into the index?
 - What can you say about the last entry that was inserted into the index if you know that there have been no deletions from this index so far?
 - Suppose you are told that there have been no deletions from this index so far. What can you say about the last entry whose insertion into the index caused a split?
 - Show the index after inserting an entry with hash value 68.
 - Show the index after inserting entries with hash values 17 and 69 into the original tree.
 - Show the index after deleting the entry with hash value 21 into the original tree. (Assume that the full deletion algorithm is used.)
 - Show the index after deleting the entry with hash value 10 into the original tree. Is a merge triggered by this deletion? If not, explain why. (Assume that the full deletion algorithm is used.)

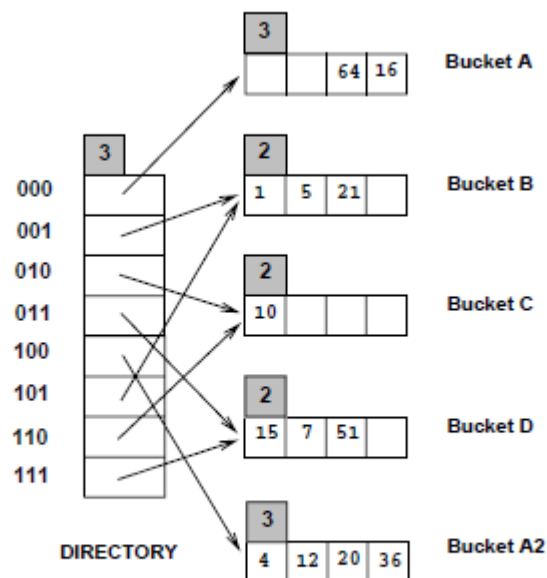


Figure 11.1 Figure for Exercise 11.1

Απαντήσεις:

- i. Μπορεί να είναι οποιοδήποτε καταχώρηση. Φυσικά πάντα μπορούμε να βρούμε μια συνάχια μεταξύ των εισαγωγών και των διαγραφών παίρνοντας για παράδειγμα μια συγκεκριμένη τιμή μεταξύ των τιμών που φαίνονται να έχουν εισαχθεί τελευταίες σε κάθε bucket.

Πάρτε για παράδειγμα την τελευταία καταχώρηση με τιμή 16 και προσπαθήστε να σκεφτείτε την εξής ακολουθία:

1 5 21 10 15 7 51 4 12 36 64 8 24 56 16 56D 24D 8D

- ii. Αποκλείετε η τελευταία εισαγωγή καταχώρησης να είχε σαν αποτέλεσμα το χώρισμα του hash αφού το άθροισμα των καταχωρήσεων στα buckets A και A2 είναι 6. Αν η τελευταία τιμή είχε προκαλέσει το περεταίρω χώρισμα του hash το άθροισμα των τιμών θα έπρεπε να ήταν 5.
- iii. Η τελευταία καταχώρηση που προκάλεσε χωρισμό του hash και αύξηση του global depth δεν θα μπορούσε να ήταν από το bucket C. Τα buckets B και C είτε τα buckets C και D θα μπορούσαν να είχαν προκαλέσει επέκταση του hash και αύξηση του global depth αλλά επειδή ξέρουμε ότι δεν υπήρξε διαγραφή καμίας καταχώρησης και το άθροισμα των καταχωρήσεων της ένωσης και των δύο buckets είναι 4, αποκλείουμε και αυτό το ενδεχόμενο. Τέλος ο συνδυασμός A και A2 καθώς και B με D θα μπορούσαν να είχαν προκαλέσει επέκταση του hash και αύξηση του global depth. Αλλά παίρνοντας σαν δεδομένο ότι το starting global depth ήταν 2, τότε επέκταση του hash και αύξηση του global depth θα μπορούσε να προκαλέσει μόνο ο συνδυασμός A και A2.
- iv. Εικόνα 11.2

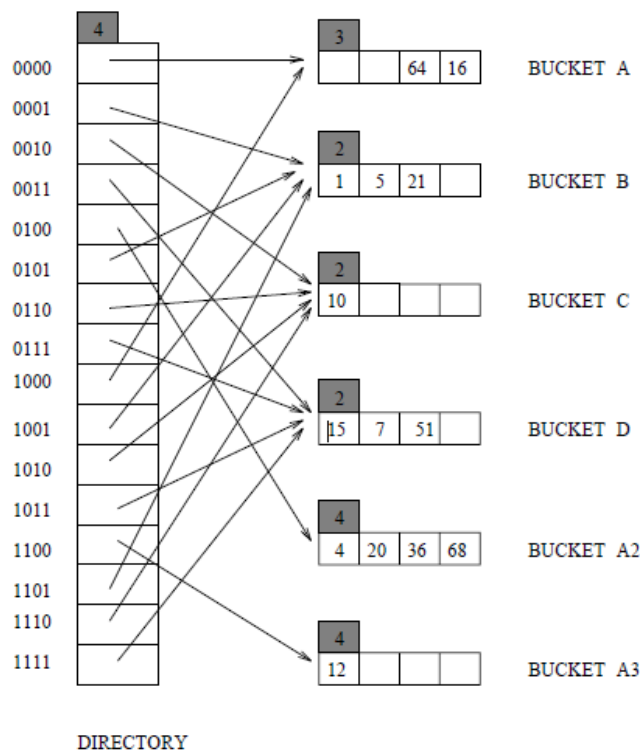


Figure 11.2

v. Εικόνα 11.3

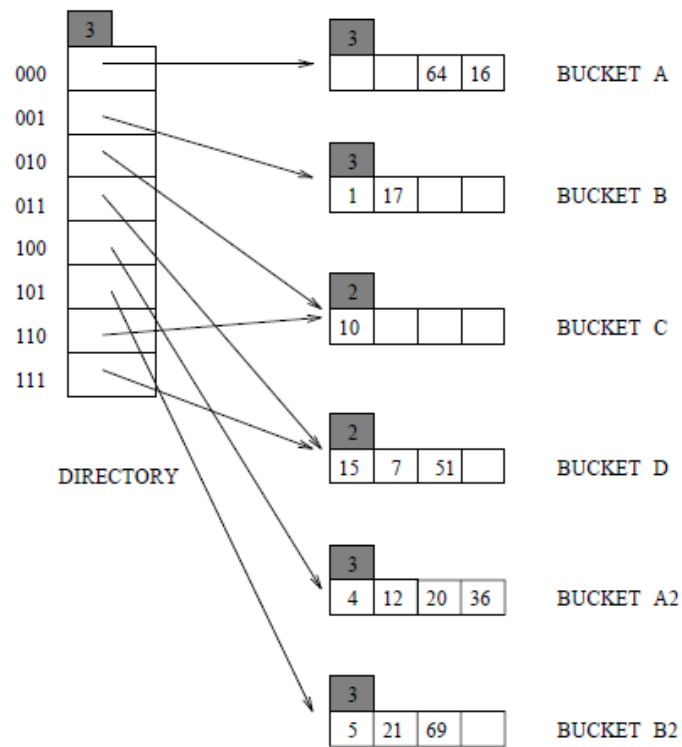


Figure 11.3

vi. Εικόνα 11.4

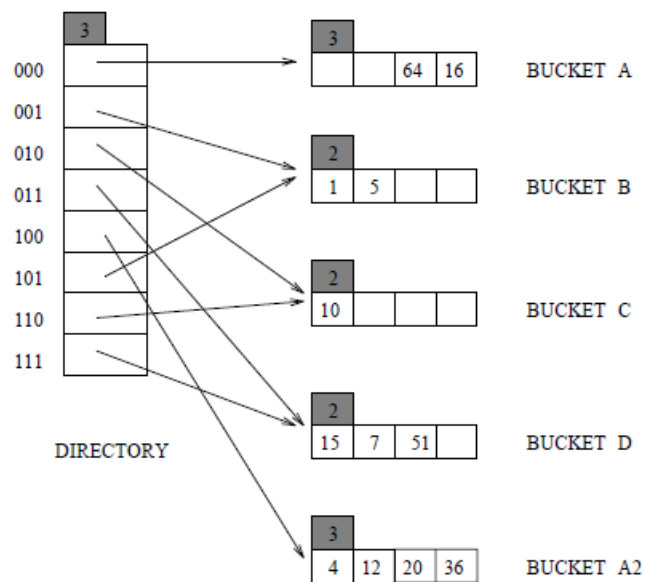


Figure 11.4

vii. Η διαγραφή της καταχώρηση με τιμή 10 δεν μπορεί να προκαλέσει την ένωση κάποιων buckets έστω και αν είναι η τελευταία καταχώρηση στο Bucket C γιατί το Bucket C είναι primary page και πρέπει να παραμείνει σαν place holder. Ακόμα όπως βλέπουμε τα directories 010 και 110 δείχνουν στο ίδιο bucket, Bucket C και αυτό είναι ακόμη μια ένδειξη πως δεν μπορεί να γίνει περεταίρω ένωση των buckets. Εικόνα 11.5

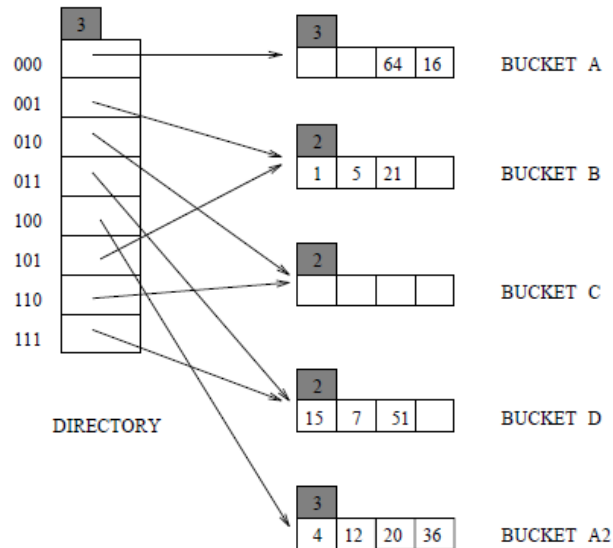


Figure 11.5