

Λύσεις Πρώτης Σειράς Ασκήσεων

Λύση Άσκησης 1

1. Ο δίσκος έχει 10.000 κυλίνδρους, δηλαδή 10.000 ίχνη (tracks) ανά επιφάνεια (surface)
Συνολικός αριθμός ιχνών = Αριθμός ιχνών ανά επιφάνεια * Αριθμός επιφανειών
 $= 10.000 * 10 = \mathbf{100.000}$

Κάθε ίχνος έχει 1.000 τομείς (sectors)
Συνολικός αριθμός τομέων = Συνολικός αριθμός ιχνών * 1.000 = $100.000 * 1.000 =$
100.000.000

Κάθε τομέας χωράει 512 bytes.
Συνολική χωρητικότητα δίσκου = Συνολικός αριθμός τομέων * 512 = $100.000.000 * 512 =$
 $51.200.000.000 / 2^{30} = \mathbf{47,6837\ GB}$.

2. Η ταχύτητα περιστροφής του δίσκου είναι 10.000 rpm (rotations per minute). Ο χρόνος που απαιτείται για να κάνει ο δίσκος μια ολόκληρη περιστροφή είναι: $60/10.000 = 0,006\ sec = 6\ msec$. Οπότε η μέση καθυστέρηση περιστροφής, δηλαδή ο χρόνος που απαιτείται για να κάνει ο δίσκος μισή περιστροφή είναι:

Μέση Καθυστέρηση Περιστροφής $= 6/2 = \mathbf{3\ msec}$

3. Εφόσον το μέγεθος του τομέα είναι 512 bytes και το μέγεθος του block είναι 4096 bytes, ένα block αποτελείται από $4096/512 = 8$ τομείς.

Αριθμός τομέων ανά block = $4096/512 = \mathbf{8}$

Σε μία περιστροφή, δηλαδή σε 6 msec, ο δίσκος διαβάζει 1.000 τομείς άρα ο χρόνος μεταφοράς ενός block είναι:

Χρόνος μεταφοράς ενός block = $6/1000 * 8 = \mathbf{0,048\ msec}$

4. Για να διαβάσουμε το ολόκληρο το αρχείο πρέπει να διαβάσουμε 1.000.000 blocks. Οπότε ο χρόνος για την ανάγνωση του αρχείου υπολογίζεται ως εξής:

Χρόνος ανάγνωσης του αρχείου = Μέσος Χρόνος Μετακίνησης Κεφαλής + Μέση Καθυστέρηση Περιστροφής + Χρόνος Μεταφοράς = $8 + 3 + 1.000.000 * 0,048 = 48.011\ msec =$
48,011 sec

5. Χρόνος ανάγνωσης N εγγραφών = $N * 4 * (8 + 3 + 0,048) = 44,192 * N\ msec = \mathbf{0,044192 * N\ sec}$

Για κάθε εγγραφή πρέπει να διαβαστούν 4 blocks. Τρία blocks για την διάσχιση του ευρετηρίου, το οποίο έχει την μορφή B+ δέντρου με τρία επίπεδα και το block που περιέχει την εγγραφή.

Λύση Άσκησης 2

1. UPDATE Employees SET age = age + 1

Η ύπαρξη του ευρετηρίου στο πεδίο age επιβραδύνει την εκτέλεση της εντολής ενημέρωσης. Εκτός από τις σελίδες που περιέχουν τα δεδομένα πρέπει να ενημερωθούν και οι σελίδες που περιέχουν τις εγγραφές του ευρετηρίου.

2. UPDATE Employees SET salary = salary * 1.10 WHERE empid >=1 and empid <=10

Το ευρετήριο συστάδων στο πεδίο empid θα αξιοποιηθεί ώστε να εντοπιστούν οι εγγραφές που πρέπει να ενημερωθούν. Η εκτέλεση της εντολής θα επιταχυνθεί διότι θα διαβαστούν λιγότερες σελίδες.

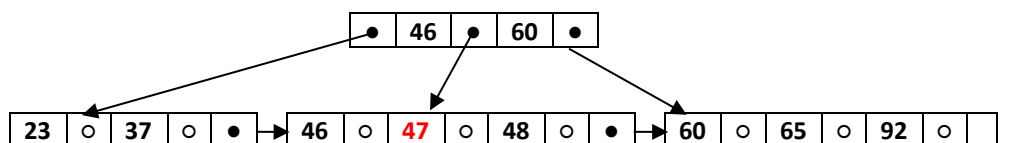
3. UPDATE Employees SET salary=salary * 1.10 WHERE departmentid = 10

Η εκτέλεση της εντολής ενημέρωσης δεν επηρεάζεται από την ύπαρξη των ευρετηρίων διότι τα ευρετήρια δεν αφορούν κάποιο από τα πεδία salary, departmentid.

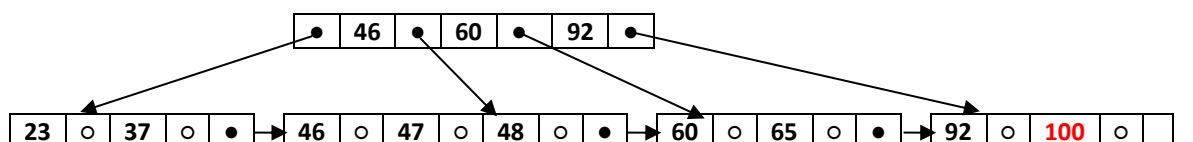
Λύση Άσκησης 3

- 1.

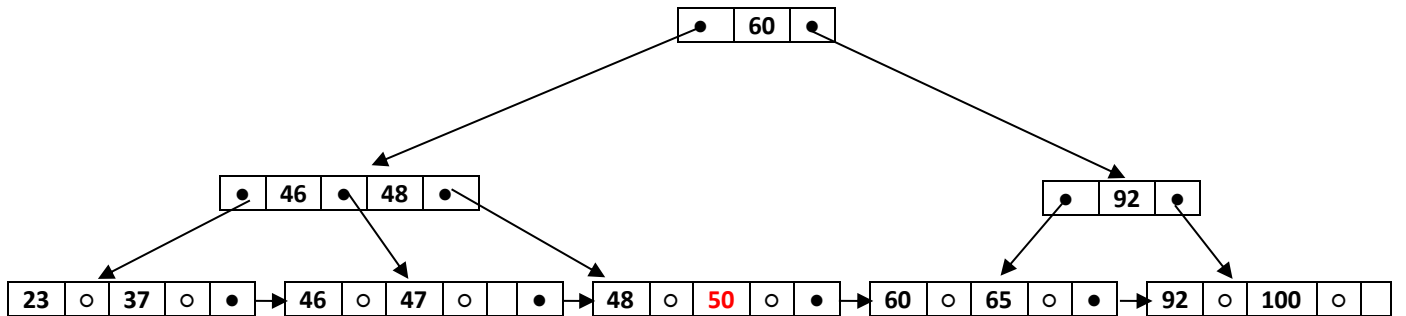
Εισαγωγή τιμής 47 : Η τιμή 47 τοποθετείται στον δεύτερο κόμβο-φύλλο διότι είναι μεγαλύτερη του 46 και μικρότερη του 60. Εφόσον $n=3$ και ο κόμβος-φύλλο περιέχει δύο τιμές χωράει και η τιμή 47 η οποία θα τοποθετηθεί ανάμεσα από τις τιμές 46 και 48 ($46 < 47 < 48$).



Εισαγωγή τιμής 100 : Η τιμή 100 τοποθετείται δεξιά στον τρίτο κόμβο-φύλλο διότι είναι μεγαλύτερη του 92. Επειδή όμως η χωρητικότητα του κόμβου-φύλλου είναι 3 ($n=3$) η εισαγωγή της τιμής 100 θα προκαλέσει υπερχείλιση. Θα δημιουργηθεί ένας νέος κόμβος στον οποίο θα καταχωρηθούν οι τιμές 92 και 100, ενώ ταυτόχρονα η τιμή 92 θα αντιγραφεί στον κόμβο ρίζα με έναν δείκτη προς το νέο κόμβο.



Εισαγωγή τιμής 50: Η τιμή 50 τοποθετείται στον δεύτερο από αριστερά κόμβο-φύλλο διότι είναι μεγαλύτερη του 48 και μικρότερη του 60. Ο κόμβος-φύλλο θα υπερχειλίσει οπότε και θα δημιουργηθεί ένας νέος κόμβος στον οποίο θα καταχωρηθούν οι τιμές 48 και 50, ενώ ταυτόχρονα η τιμή 50 θα αντιγραφεί στον κόμβο ρίζα μαζί με έναν δείκτη προς το νέο κόμβο. Επειδή η ρίζα περιέχει ήδη τρεις τιμές θα υπερχειλίσει. Ως αποτέλεσμα θα δημιουργηθεί ένας νέος κόμβος ο οποίος θα πάρει τις τιμές 60 και 92. Στον νέο κόμβο χρειαζόμαστε μία τιμή και δύο δείκτες που θα δείχνουν στα φύλλα, οπότε η τιμή 60 μετακινείται ένα επίπεδο πάνω και δημιουργεί την ρίζα του δέντρου.



2.

α) Αν θεωρήσουμε ότι το κλειδί αναζήτησης είναι μοναδικό τότε θα επισκεφτούμε 5 κόμβους:

[60] --> [46,48]-->[46,47]-->[48,50]-->[60,65]

β) Αν θεωρήσουμε ότι το κλειδί αναζήτησης δεν είναι μοναδικό (υπάρχουν διπλότυπα) τότε θα επισκεφτούμε 6 κόμβους:

[60] --> [46,48]-->[46,47]-->[48,50]-->[60,65]--> [92,100]

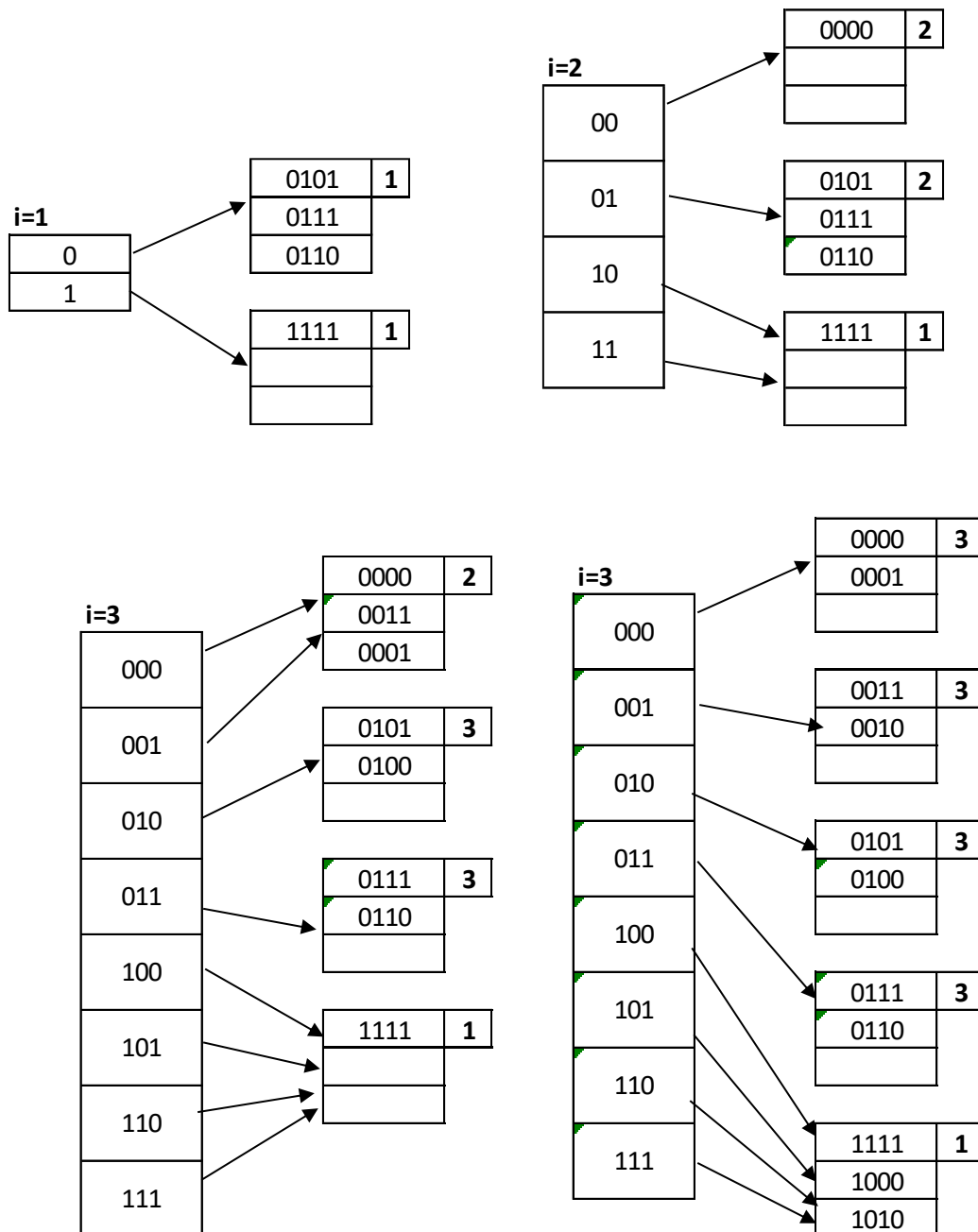
Λύση Άσκησης 4

Χρειαζόμαστε **200** μπλοκ (1000/5) για την αποθήκευση της σχέσης R. Ένα αραιό ευρετήριο θα έχει ένα δείκτη για κάθε ένα από τα 200 block. Στα φύλλα έχουμε 10 δείκτες προς τα δεδομένα και έναν δείκτη προς το επόμενο φύλλο, δηλαδή χρειαζόμαστε **20** block για να αποθηκεύσουμε τα φύλλα του B+ ευρετηρίου. Στο αμέσως παραπάνω επίπεδο του δέντρου μπορούμε να δείξουμε από κάθε κόμβο σε 11 κόμβους φύλλα, οπότε θα χρειαστούμε επιπλέον **2** κόμβους, δηλαδή άλλα δύο block. Τέλος θα χρειαστούμε άλλο ένα block για την ρίζα του δέντρου.

Σύνολο: 200+20+2+1=223 blocks.

Λύση Άσκησης 5

1.



2.

Το ευρετήριο διπλασιάζεται όταν το νέο τοπικό βάθος ξεπεράσει το τρέχον ολικό βάθος. Παρατηρούμε ότι υπάρχουν 4 κάδοι με τοπικό βάθος 3 όσο δηλαδή και το ολικό. Αν τοποθετήσουμε 2 διαδοχικές τιμές σε κάποιον από αυτούς τους κάδους τότε το ολικό βάθος του ευρετηρίου θα αυξηθεί κατά 1. Με άλλα λόγια το ευρετήριο θα διπλασιαστεί. Π.χ. η εισαγωγή των τιμών 0001 και 0000 στον πρώτο κάδο θα οδηγήσει σε αύξηση του τοπικού βάθους κατά 1. Ο πρώτος κάδος θα διασπαστεί σε δύο κάδους και το ευρετήριο θα διπλασιαστεί.

Λύση Άσκησης 6

Θα πρέπει όλες οι N εγγραφές που περιέχει ο κάδος καθώς και η νέα εγγραφή που θα προκαλέσει την υπερχείλιση να έχουν το ίδιο bit στην θέση $i=j+1$ όπου j το τοπικό βάθος του κάδου πριν από την διάσπαση: $2 \cdot (1/2)^{N+1} = (1/2)^N$.

Επεξήγηση: Θα πρέπει το bit στην θέση i των $N+1$ αριθμών να είναι το ίδιο. Αν τοποθετήσουμε αυτά τα $N+1$ bits στη σειρά έχουμε έναν δυαδικό αριθμό με $N+1$ bits. Αυτός παίρνει 2^{N+1} τιμές. Για παράδειγμα:

```
00000....0
00000....1
00000...01
00000...10
...
...
....

11111...1
```

Μόνο η πρώτη και η τελευταία τιμή όπου όλα τα bits είναι ίδια (όλα 0 ή όλα 1) προκαλούν νέα διάσπαση. Αρά από τους 2^{N+1} παραπάνω συνδυασμούς θέλουμε μόνο τους 2. Αυτό συμβαίνει με πιθανότητα $2/(2^{N+1}) = (1/2)^N$.