

Συστήματα Διαχείρισης και Ανάλυσης Δεδομένων Διδάσκων: Ιωάννης Κωτίδης

Εαρινό εξάμηνο 2025-2026

Λύση άσκησης 1

1. Υπολογισμός των blocks που καταλαμβάνει το αρχείο δεδομένων.

Μέγεθος εγγραφής = $8+12+8+8+4=40$ bytes.

Σε ένα block χωράνε: $\text{floor}(4096/40)=\text{floor}(102,4) = 102$ εγγραφές.

Συνεπώς το αρχείο καταλαμβάνει: $\text{ceiling}(3000000/102)=\text{ceiling}(29411,76)=29412$ blocks.

2. Υπολογισμός Χρόνου μεταφοράς ενός block.

Σε μία περιστροφή ο δίσκος διαβάζει 256 τομείς. Δεδομένου ότι ένα block καταλαμβάνει $4096/512 = 8$ τομείς, σε μια στροφή ο δίσκος διαβάζει $256/8=32$ blocks. Ο χρόνος μιας πλήρους περιστροφής είναι: $(1/10000)*60=0,006$ sec = **6ms**. Συνεπώς:

Χρόνος μεταφοράς block (tb) = $6/32=0,1875$ ms

Χρόνος μισής περιστροφής (trd) = $6/2 = 3$ ms

Μέσος χρόνος μετακίνησης κεφαλής(ts) = **12ms** (Από τα δεδομένα της άσκησης)

Χρόνος σειριακής ανάγνωσης του αρχείου = $ts+trd+tb*(29412)=12 + 3 + (0,1875*29412)$
= **5529,75ms**

3. Υπολογισμός Χρόνου ανάγνωσης N εγγραφών με τυχαία προσπέλαση

Για να διαβάσουμε μια εγγραφή με τυχαία προσπέλαση πρέπει να διαβάσουμε το block στο οποίο βρίσκεται η εγγραφή. Ο χρόνος για την ανάγνωση ενός block είναι:

Χρόνος Ανάγνωσης block = $ts+trd+tb=12\text{ms}+3\text{ms}+0,1875=15,1875\text{ms}$.

Για να διαβάσουμε N εγγραφές απαιτούνται **N*15,1875ms**.

4. Σύγκριση

Για να συμφέρει η τυχαία προσπέλαση θα πρέπει:

$N*15,1875 < 5529,75 \rightarrow N < 5529,75 / 15,1875 \rightarrow N < 364,1 \leq 364$ εγγραφές.

Συμπέρασμα: Η σωστή επιλογή είναι το B. Ο ισχυρισμός ισχύει για $N \leq 364$ εγγραφές.

Λύση Άσκησης 2

a.

Για την αποθήκευση της σχέσης R απαιτούνται **300** blocks (1800/6).

Ένα **αραιό ευρετήριο** θα έχει ένα δείκτη για κάθε ένα από τα 300 blocks. Στα φύλλα έχουμε 9 δείκτες προς τα δεδομένα και έναν δείκτη προς το επόμενο φύλλο, δηλαδή χρειαζόμαστε $\text{ceiling}(300/9)=34$ blocks για να αποθηκεύσουμε τα φύλλα του β+ δέντρου. Στο αμέσως παραπάνω επίπεδο του δέντρου μπορούμε να δείξουμε από κάθε κόμβο σε 10 κόμβους φύλλα, οπότε θα χρειαστούμε επιπλέον $\text{ceiling}(34/10)=4$ κόμβους, δηλαδή άλλα **4** blocks. Τέλος θα χρειαστούμε άλλο **1** block για την ρίζα του δέντρου.

Συνολικά χρειαζόμαστε $34 + 4 + 1 = 39$ blocks

Ένα **πυκνό ευρετήριο** θα έχει ένα δείκτη για κάθε μία εγγραφή. Στα φύλλα έχουμε 9 δείκτες προς τα δεδομένα και έναν δείκτη προς το επόμενο φύλλο, δηλαδή χρειαζόμαστε $1800/9=200$ blocks για να αποθηκεύσουμε τα φύλλα του β+ δέντρου. Στο αμέσως παραπάνω επίπεδο του δέντρου μπορούμε να δείξουμε από κάθε κόμβο σε 10 κόμβους φύλλα, οπότε θα χρειαστούμε επιπλέον $200/10=20$ κόμβους, δηλαδή άλλα **20** blocks. Με την ίδια λογική στο αμέσως παραπάνω επίπεδο θα χρειαστούμε $20/10=2$ blocks. Τέλος θα χρειαστούμε **1** block για την ρίζα του δέντρου.

Συνολικά χρειαζόμαστε $200 + 20 + 2 + 1 = 223$ blocks

b.

Παρατηρούμε ότι το β+ δέντρο για το αραιό ευρετήριο έχει ύψος 3 και το β+ δέντρο για το πυκνό ευρετήριο έχει ύψος 4. Αυτό σημαίνει ότι για να ανακτήσουμε μια εγγραφή χρησιμοποιώντας το **αραιό** ευρετήριο πρέπει να προσπελάσουμε:

Την ρίζα του ευρετηρίου → 1 block

Έναν κόμβο στο ενδιάμεσο επίπεδο → 1 block

Έναν κόμβο στο επίπεδο των φύλλων → 1 block

Το block που περιέχει τα δεδομένα της εγγραφής → 1 block

Σύνολο 4 blocks

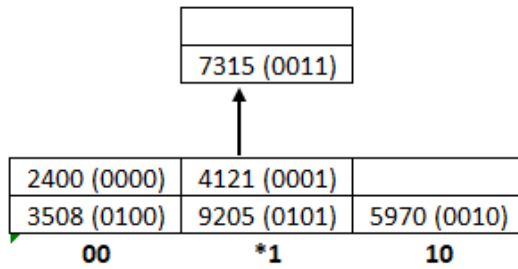
Με την ίδια λογική χρησιμοποιώντας το πυκνό ευρετήριο πρέπει να προσπελάσουμε έναν κόμβο παραπάνω διότι σε αυτή την περίπτωση το β+ δέντρο έχει 4 επίπεδα.

Σύνολο 5 blocks.

Συνεπώς το αραιό ευρετήριο είναι καταλληλότερο για την εκτέλεση του επερωτήματος.

Λύση Άσκησης 3

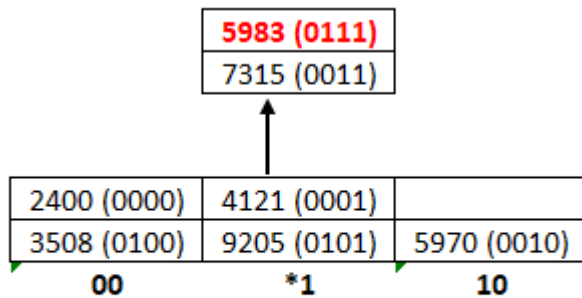
a.



$m=2$ (10) $i=2$ $U=75\%$

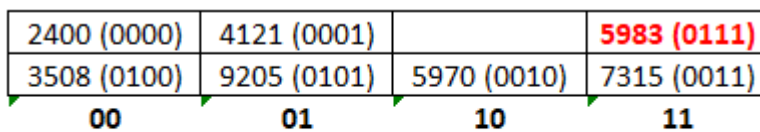
b.

1 Εισαγωγή **5983 (0111)**



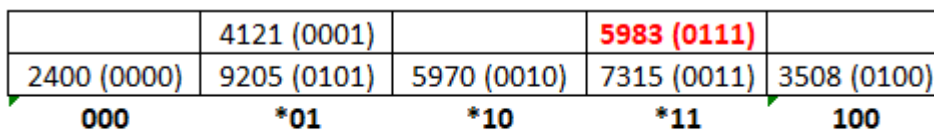
$m=10$ (2) $i=2$ $U=87,5\%$

Utilization > 80%, οπότε $m=11$ (3)



$m=11$ (3) $i=2$ $U=87,5\%$

Utilization > 80%, το i αυξάνεται σε 3 και το $m=100$ (4)



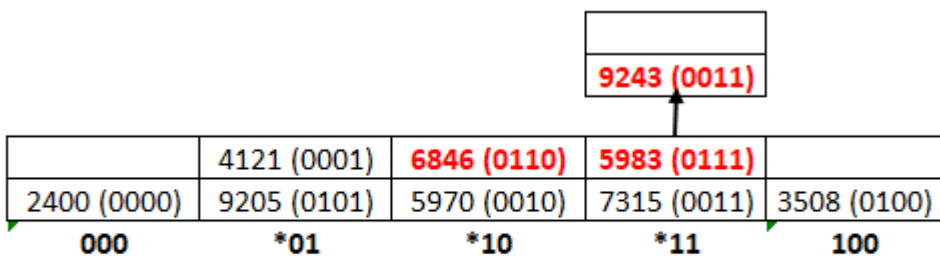
$m=100$ (4) $i=3$ $U=70\%$

2 Εισαγωγή **6846 (0110)**

	4121 (0001)	6846 (0110)	5983 (0111)	
2400 (0000)	9205 (0101)	5970 (0010)	7315 (0011)	3508 (0100)
000	*01	*10	*11	100

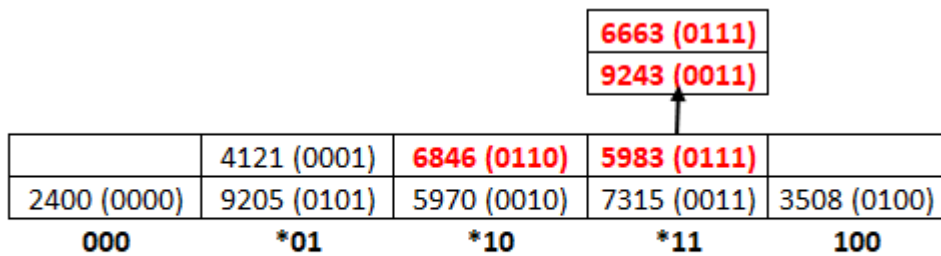
m=100 (4) i=3 U=80%

3 Εισαγωγή **9243 (0011)**



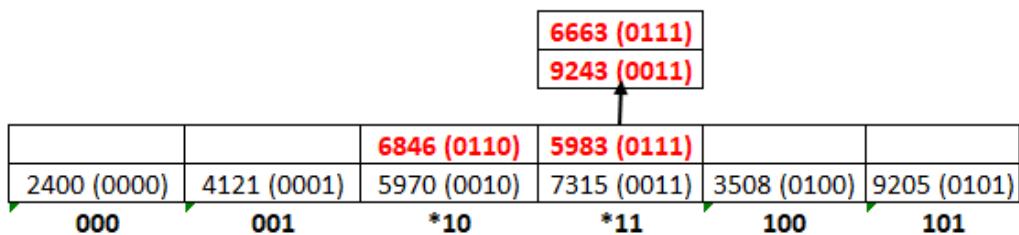
m=100 (4) i=3 U=75%

4 Εισαγωγή **6663 (0111)**



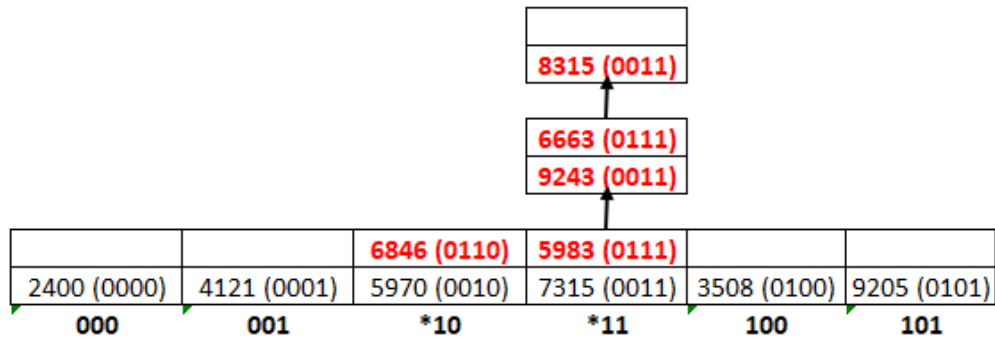
m=100 (4) i=3 U=83,3%

Utilization > 80%, οπότε m=101 (5)



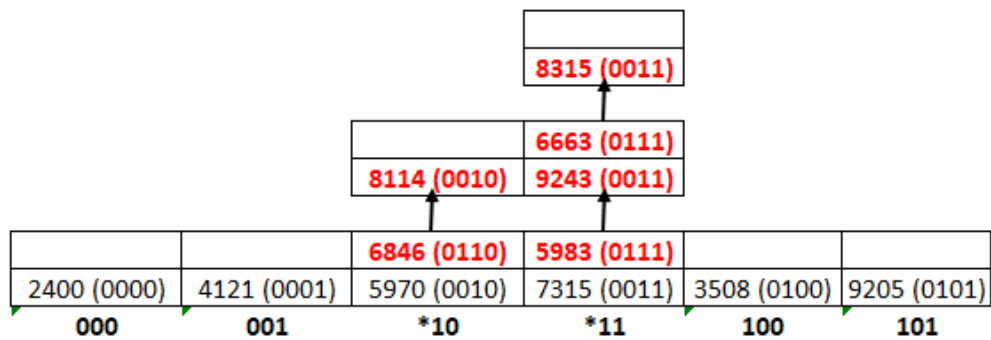
m=101 (5) i=3 U=71,4%

5 Εισαγωγή **8315 (0011)**



m=101 (5) i=3 U=68,75%

6 Εισαγωγή **8114 (0010)**



m=101 (5) i=3 U=66,67%

c.

Ο μέσος αριθμός προσπελάσεων είναι:

- 8 εγγραφές απαιτούν 1 προσπέλαση
- 3 εγγραφές απαιτούν 2 προσπελάσεις
- 1 εγγραφή απαιτεί 3 προσπελάσεις

Συνεπώς έχουμε: $(8 * (1/12)) + (3 * (2/12)) + (1 * (3/12)) = 1,41$

d.

Μέσος αριθμός Προσπελάσεων όταν το κλειδί δεν υπάρχει

Αν αναζητούμε τιμές που δεν υπάρχουν τότε εξετάζοντας τα 3 τελευταία bits το κόστος αναζήτησης θα είναι:

000 → 1

001 → 1

010 → 2

011 → 3

100 → 1

101 → 1

110 → 2

111 → 3

Συνεπώς έχουμε: $(4 * (1/8)) + (2 * (2/8)) + (2 * (3/8)) = 14/8=1,75$