Τζένη Μπολένα
3170117

*Για τα διαγράμματα χρησιμοποιήθηκε το εργαλείο https://excalidraw.com

Πρώτη Σειρά Ασκήσεων

Άσκηση 1

a)

- Ένα ίχνος αποτελείται από 256 τομείς. Ένας τομέας αποτελείται από 4096 bytes. Άρα η χωρητικότητα ενός ίχνος είναι:
  256 * 4096 bytes= 1048576 bytes = 1 MB
- Μια επιφάνεια έχει 65536 tracks. Το ένα track έχει χωρητικότητα 1048576 bytes. Άρα η χωρητικότητα μιας επιφάνειας είναι:
  65536 * 1048576 bytes = 68719476736 bytes = 65536 MB = 64GB
- Ο δίσκος αποτελείται από 8 πλακέτες διπλής όψης, άρα από 16 επιφάνειες. Άρα η χωρητικότητα του δίσκου είναι:
  16 * 68719476736 bytes = 1099511627776 bytes = 1048576 MB = 1024 GB = 1 TB

b) Ο αριθμός των κυλίνδρων του δίσκου είναι όσο και τα tracks μιας επιφάνειας. Άρα έχουμε **65536 κυλίνδρους** στον δίσκο.

c)

- Κατά μέσο όρο θα κάνω μισή περιστροφή. Η ταχύτητα περιστροφής του δίσκου είναι 7200 rpm. Οπότε η μέση καθυστέρηση περιστροφής είναι:
  60 sec 7200 rotations
  X ½ rotation

  X = 30 sec/ 7200 ⇔ X = 0.00416 sec = 4.159 ms

- Η μέγιστη καθυστέρηση επιστροφής θα συμβεί όταν κάνουμε ένα ολόκληρο rotation. Οπότε η μέγιστη καθυστέρηση περιστροφής είναι:
  60 sec 7200 rotations
  X 1 rotation

  X = 60 sec/ 7200 ⇔ X = 0.00833 sec 8.33 ms

d) Σε μια περιστροφή διαβάζω ένα ίχνος άρα 256 τομείς. Το transfer time για ένα track είναι **60 / 7200 = 0.00833 sec = 8.33 ms**. Ένα ίχνος έχει χωρητικότητα **1 MB.** Άρα ο ρυθμός μεταφοράς(transfer rate) είναι:
transfer rate = data to be transferred / transfer time ⇔
Transfer rate = 1MB / 0.00833 sec = 120MB/sec

Άσκηση 2

Ας ξεκινήσουμε με το πόσες σελίδες/blocks καταλαμβάνει όλη η σχέση R.Έχουμε N εγγραφές με 5 γνωρίσματα. Η κάθε σελίδα έχει μέγεθος 1024 bytes. Άρα ο αριθμός σελίδων/ block που απαιτεί η σχέση R για αποθήκευση είναι:

N * (10 + 50 + 30 + 18 + 20) bytes / 1024 bytes = 128 * N / 1024

a) Σε ένα πυκνό ευρετήριο για το πρωτεύον κλειδί, σε κάθε εγγραφή θα υπάρχει το πρωτεύον κλειδί (10 bytes) και ο pointer(6 bytes). Άρα ο αριθμός σελίδων/ block που απαιτείται για την αποθήκευση του πυκνού ευρετηρίου είναι:

N * (10+ 6) bytes / 1024 bytes = 16 * N / 1024

Άρα ο αριθμός των blocks για την αποθήκευση της σχέσης R και του dense index είναι συνολικά:

(**128 * N / 1024) + (16 * N / 1024) = 144 * N /1024**

b) Σε ένα αραιό ευρετήριο για το πρωτεύον κλειδί, σε κάθε εγγραφή θα υπάρχει η τιμή του πρώτου κλειδιού ανά κάθε σελίδα της σχέσης R. Άρα θα έχουμε τόσες εγγραφές στο sparse index όσο και οι σελίδες/blocks που καταλαμβάνει η σχέση R. Επομένως ο αριθμός των απαιτούμενων blocks για την αποθήκευση του sparse index είναι:

((**128 * N / 1024) * (10 + 6) bytes) / 1024 bytes =**
((**128 * N / 1024) * 16) / 1024**

Άρα ο αριθμός των blocks για την αποθήκευση της σχέσης R και του sparse index είναι συνολικά:

128 * N / 1024 + ((**128 * N / 1024) * 16) / 1024**

(*Παίρνουμε το math ceiling όταν η διαίρεση δεν είναι ακέραια όσον αφορά τον αριθμό των εγγραφών για το sparse index.)

Άσκηση 3

Υπόθεση: Τις εγγραφές ενός κόμβου τις προσπελάσουμε σειριακά, αυτό θα μπορούσε να γίνει και δυαδικά. Οπότε για εξηγήσω την άσκηση κρατάω αυτήν την υπόθεση.

Σε κάθε ένα από τα παραδείγματα αναζήτησης για να καταλήξω στην εγγραφή δεδομένων ακολουθώ το data pointer της εγγραφής του index στο φύλλο.

1) Για να βρω το 41 κάνω τα εξής βήματα:

A. Έλεγχος ρίζας με το 41. 13 < 41, άρα πάω δεξιά.
B. Έλεγχος με το 23, όπου το 41 είναι μεγαλύτερο, μετά με το 31 και μετά με το 43. Το 43 είναι μεγαλύτερο από το 41,άρα ακολουθώ τον δίκτη ανάμεσα στο 31 και 43.
C. Καταλήγω σε φύλλο όπου υπάρχει το 41, το βρίσκουμε μετά το 31 και 37.

2) Για να βρω το 40 κάνω τα εξής βήματα:

A. Έλεγχος ρίζας με το 40. 13 < 40, άρα πάω δεξιά.
B. Έλεγχος με το 23, όπου το 40 είναι μεγαλύτερο, μετά με το 31 και μετά με το 43. Το 43 είναι μεγαλύτερο από το 40,άρα ακολουθώ τον δίκτη ανάμεσα στο 31 και 43.
C. Καταλήγω σε φύλλο όπου αν υπάρχει το 40 θα βρίσκεται εκεί. Συγκρίνω με 31, μετά 37, μετά ακολουθεί το 41. Παρατηρούμε ότι το 41 είναι μεγαλύτερο από το 40 και αφού δεν έχουμε βρεί το 40 μέχρι εκείνη τότε δεν υπάρχει.

3) Για να βρω τις εγγραφές με κλειδιά μικρότερα του 30 κάνω τα εξής βήματα:

A. Ελέγχω αν η ρίζα είναι μικρότερη του 30. Είναι οπότε ακολουθώ τον αριστερό pointer της ρίζας.
B. Αφου η ρίζα είναι μικρότερη του 30 θα ακολουθώ συνέχεια το πιο αριστερό pointer για να καταλήξω στο πρώτο φύλλο. Δλδ ακολουθώ το αριστερό pointer απο το 7.
C. Καταλήγω στα φύλα. Ξεκινάω από το 2 και πηγαίνω σειριακά μέχρι να βρεθεί τιμή ίση ή μεγαλύτερη του 30, στην περίπτωση μας αυτή είναι το 31. Σταματάω, και περιλαμβάνω στο αποτέλεσμα μου ότι βρήκαμε μέχρι το 31(όχι το 31.)

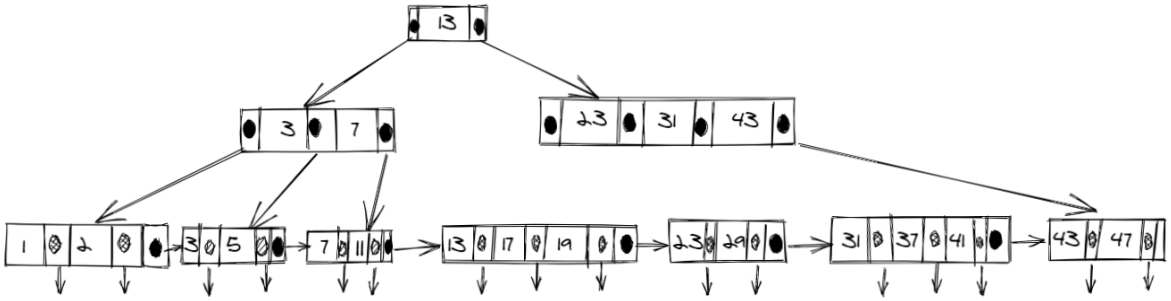
4) Για να βρω τις εγγραφές με κλειδιά στο διάστημα [20, 35] κάνω τα εξής βήματα:

A. Συγκρίνω την ρίζα με το 20, 13 , 20 άρα ακολουθώ τον δεξί pointer.
B. Πάω στο 23 όπου 20 <23 άρα ακολουθώ τον pointer αριστερά του 23.
C. Καταλήγω στα φύλλα. Θα ψάξω συριακά και θα σταματήσω όταν βρώ τιμή μεγαλύτερη του 35. Θα συμπεριλάβω ωστόσο αποτελέσματα μόνο στο διάστημα [20, 35]. Ξεκινάμε απο 13, 17, 19 που δεν τα κρατάμε στο

αποτέλεσμα μας. Μετά βρίσκουμε 23, 29, 31, τα κρατάμε. Ακολουθεί το 37, το οποίο όμως είναι μεγαλύτερο του 35 οπότε η αναζήτηση σταματάει εκεί.

(οι εικόνες φαίνονται καθαρά σε περίπτωση που κάνετε ζουμ)

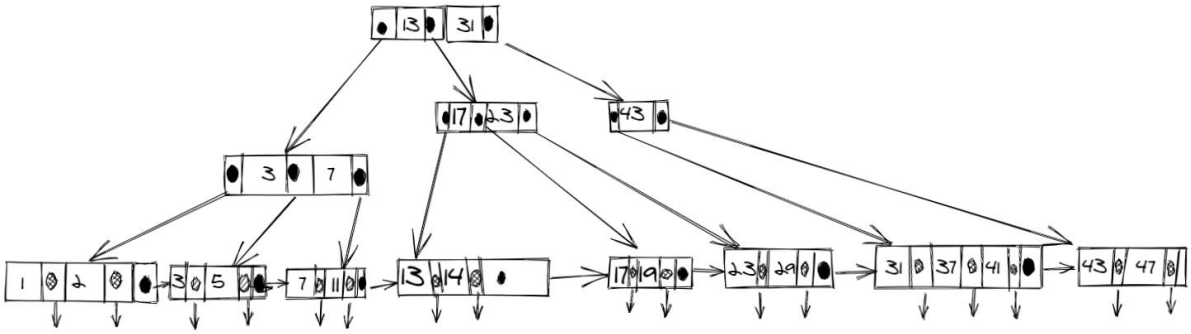
5) Εισαγωγή 1



Σχόλια: το 1 θα τοποθετηθεί αρχικά μαζί με το 2 3 5, δεν χωράει όμως οπότε το φύλλο σπάει στα δύο και το 3 θα ανέβει πάνω μαζί με το 7.

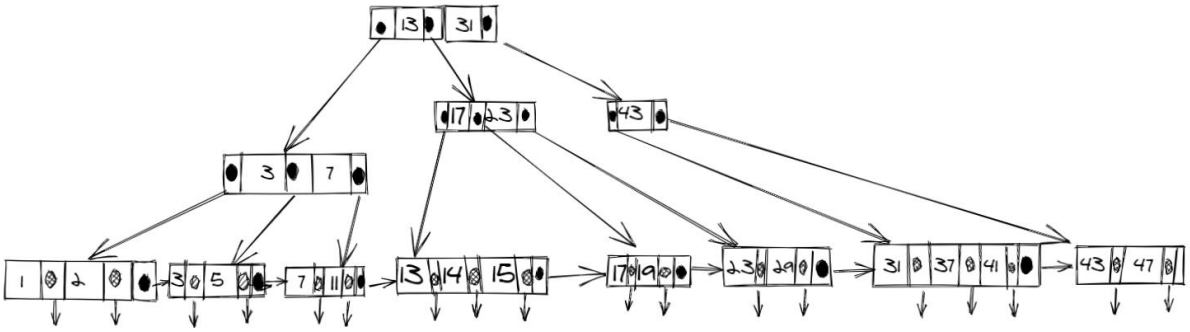
6)

Εισαγωγή 14



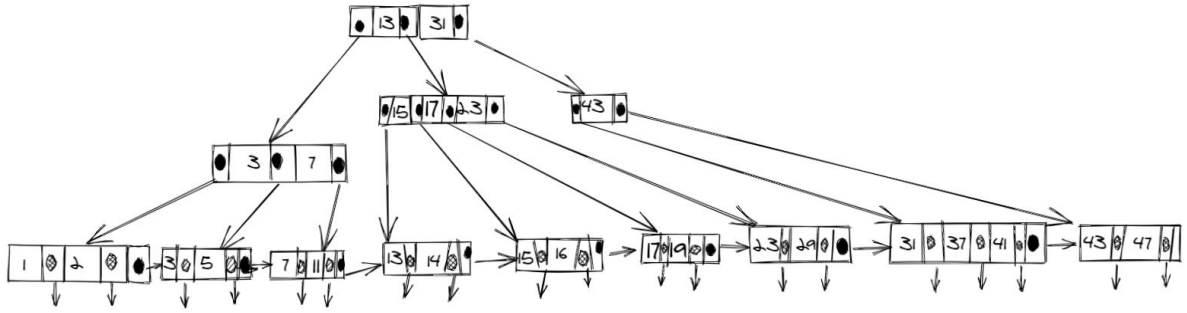
Σχόλια: το 14 θα τοποθετηθεί αρχικά μαζί με το 13 17 19, δεν χωράει όμως οπότε το φύλλο σπάει στα δύο και το 17 θα ανέβει πάνω μαζί με το 23 31 43, δεν χωράνε όλα μαζί όμως. Θα σπάσουν στα δύο και το 31 θα ανέβει στην ρίζα με το 13.

Εισαγωγη 15



Σχόλια: το 15 θα τοποθετηθεί μαζί με το 13 14, χωράει οπότε μένει εκεί.

Εισαγωγή 16(άρα και τελική μορφή δέντρου μετά από όλες τις εισαγωγές)



Σχόλια: το 16 θα τοποθετηθεί αρχικά μαζί με το 13 14 15, δεν χωράει όμως οπότε το φύλλο σπάει στα δύο και το 15 θα ανέβει πάνω μαζί με το 17 23. Έτσι σταματάει η διαδικασία και καταλήγουμε στο παραπάνω δέντρο μετά από όλες τις εισαγωγές.

Άσκηση 4

Ορίζω τις εξής μεταβλητές:

- N -> αριθμός εγγραφών σε ένα block
- Ptree -> μέγεθος pointer σε κόμβο δέντρου σε byte
- Pdata -> μέγεθος pointer προς εγγραφή δεδομένων σε byte
- κ->μέγεθος κλειδιού αναζήτησης της σχέσης R σε byte
- Block_size -.> μέγεθος block σε byte

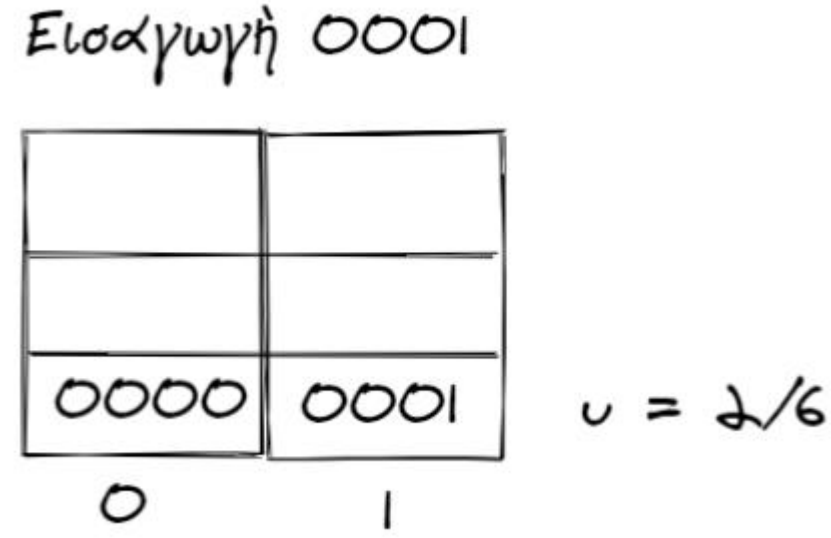
Ας ξεκινήσουμε βλέποντας πόσες εγγραφές/χωράνε σε κάθε ένα από τα επίπεδα του B+ δέντρου σε ένα block:

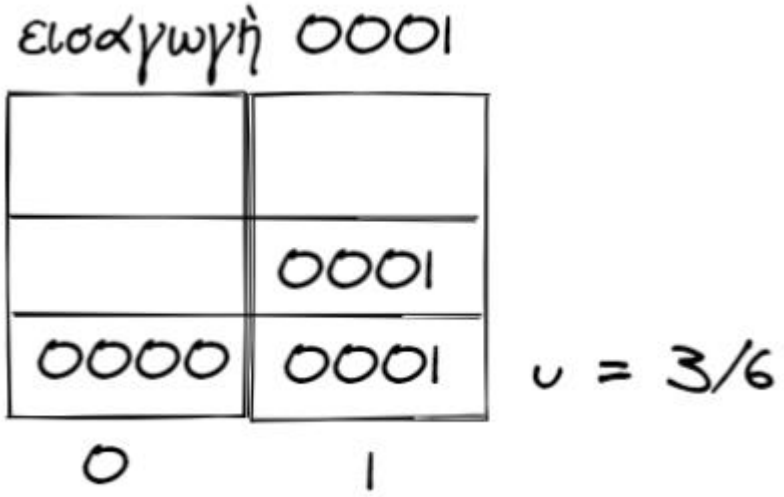
- **Ρίζα:** N*Ptree + (N-1)*k <= Block_size ⇔ N*12 + 8*N - 8 <= 2048 ⇔ 20*N <= 2056 ⇔ N <= 102.8 ⇔ **N = 102** εγγραφές στην ρίζα
- **Ενδιάμεσο επίπεδο:** N*Ptree + (N-1)*k <= Block_size ⇔ N*12 + 8*N - 8 <= 2048 ⇔ 20*N <= 2056 ⇔ N <= 102.8 ⇔ **N = 102** εγγραφές ανα block στο ενδιάμεσο επίπεδο
- **Φύλλα:** N*(k + Pdata) + Ptree <= Block_size ⇔ N*(8 + 12) + 12 <= 2048 ⇔ 20N <= 2036 ⇔ N <= 101.8 ⇔ **N = 101** εγγραφές ανα block στα φύλλα του δέντρου

Οι μέγιστες εγγραφές της σχέσης R που μπορούν να ευρετηριαστούν προκύπτουν πολλαπλασιάζοντας τον αριθμό εγγραφών στην ρίζα με τον αριθμο εγγραφών ανα block στο ενδιάμεσο επίπεδο επί τον αριθμό εγγραφών ανα block στο επίπεδο των φύλλων:

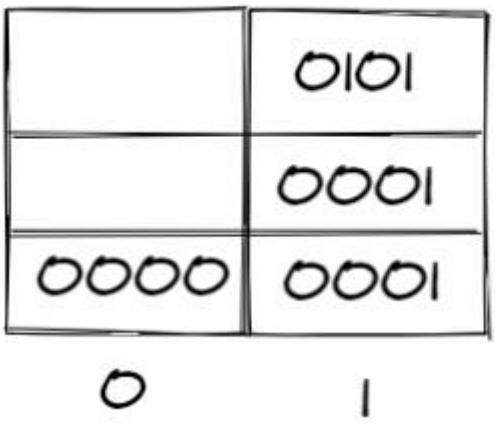
102 * 102 *101 = 1050804 εγγραφές

Άσκηση 5


Εισαγωγή 0001
0000
0001
υ = 2/6
0
1

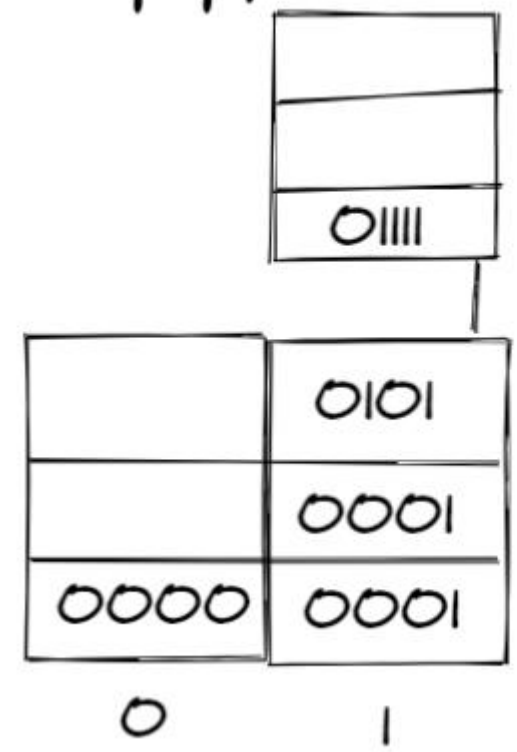

εισαγωγή 0001
0001
0000
0001
υ = 3/6
0
1

Εισαγωγή 0101

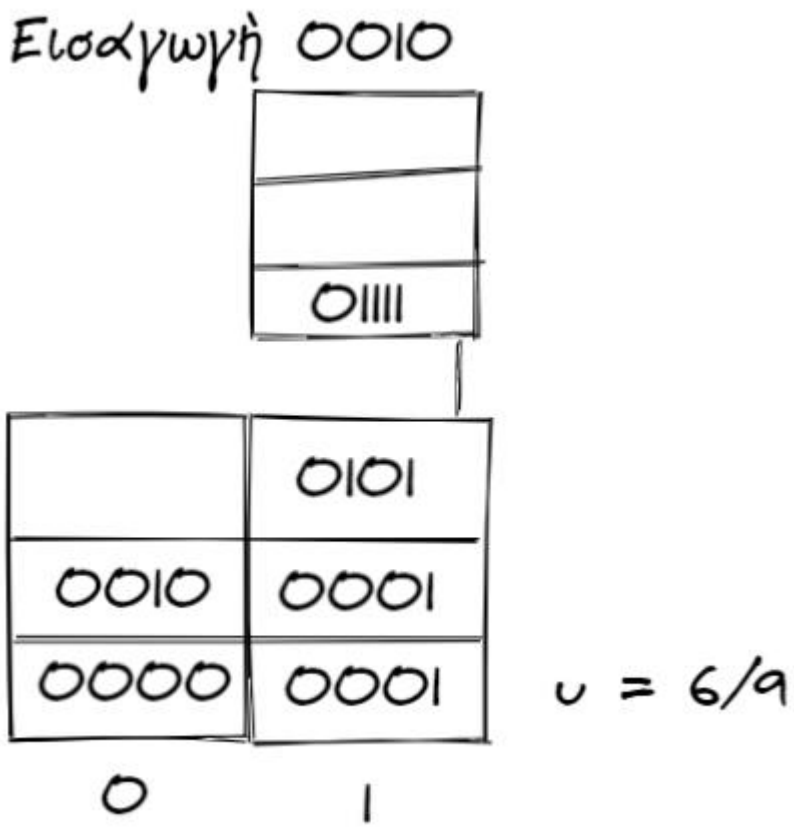


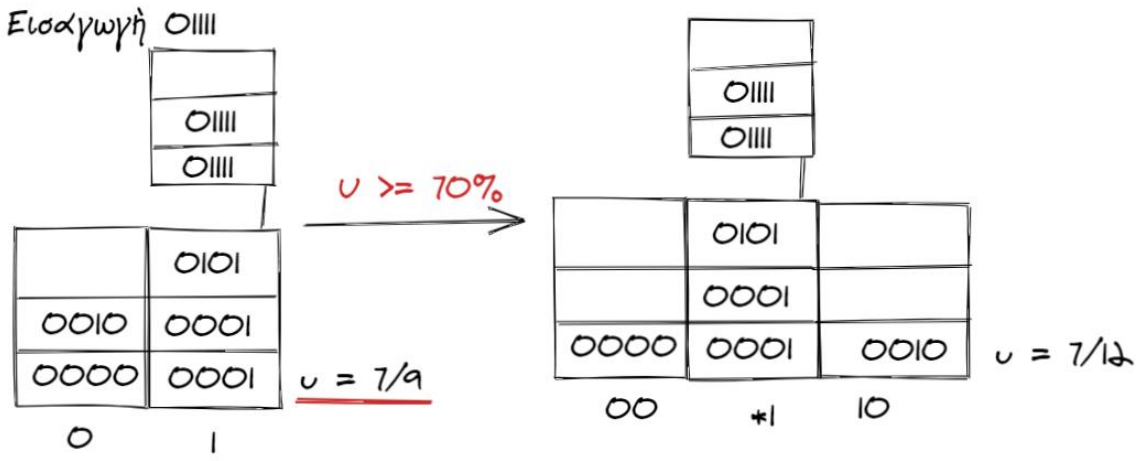
$\upsilon = 4/6$

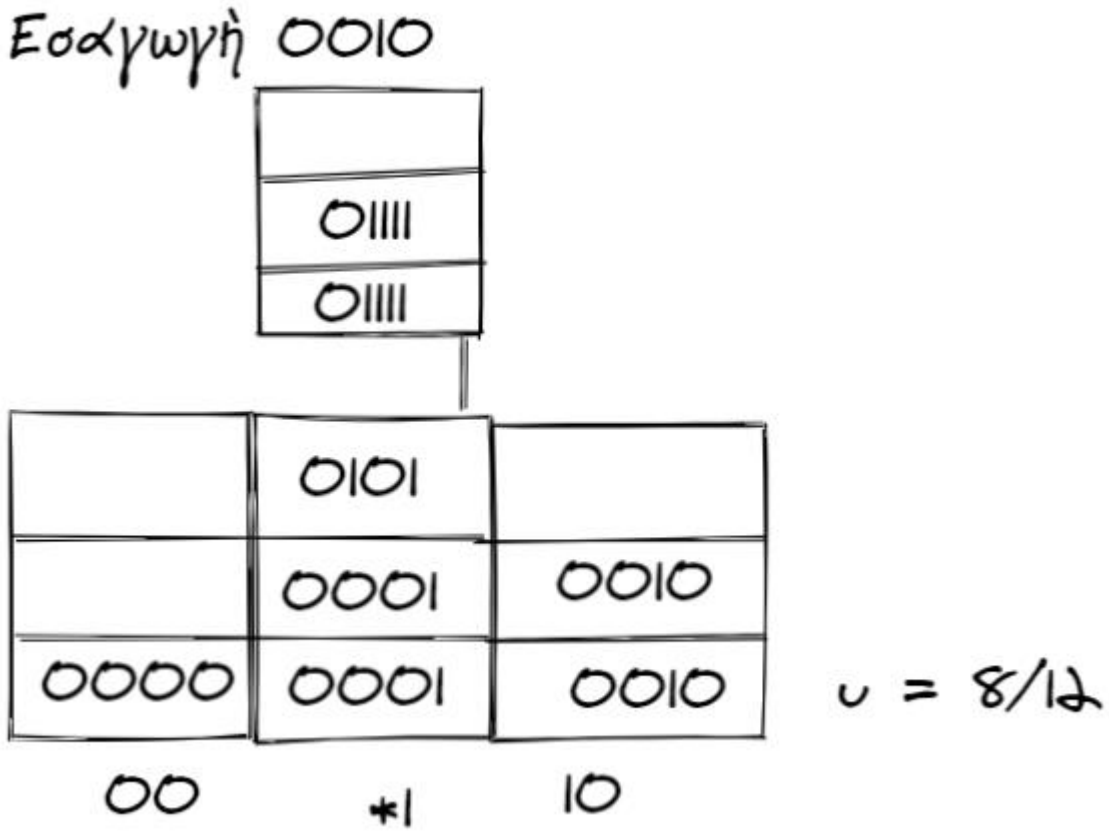
Εισαγωγή 01111

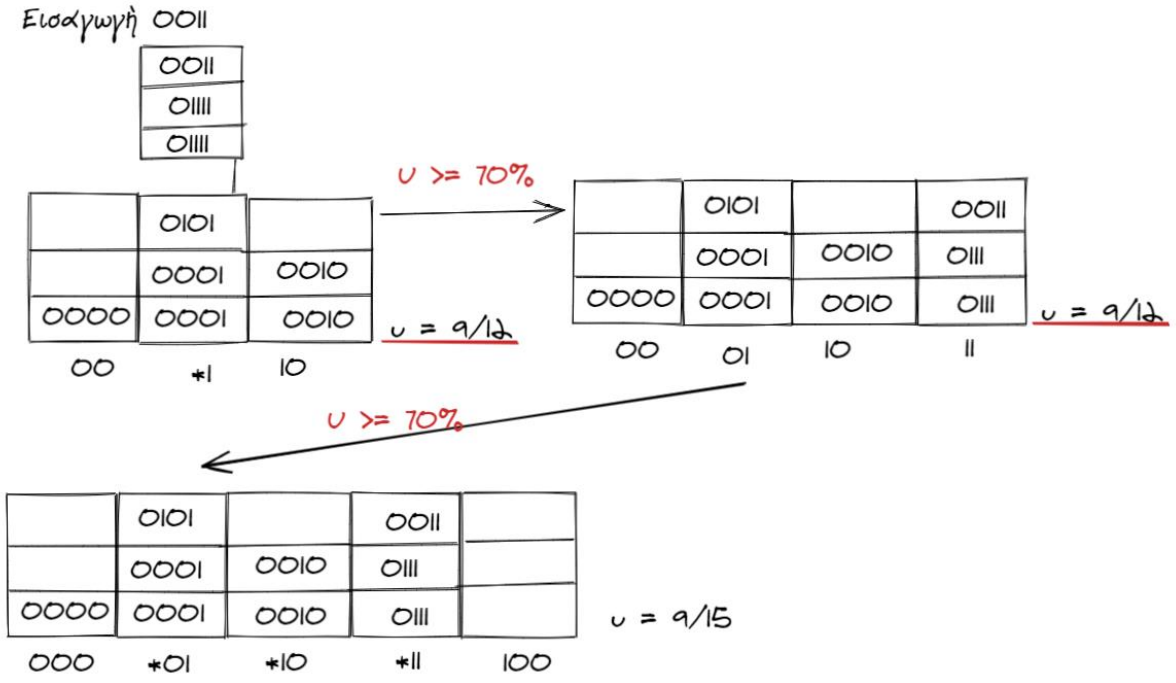


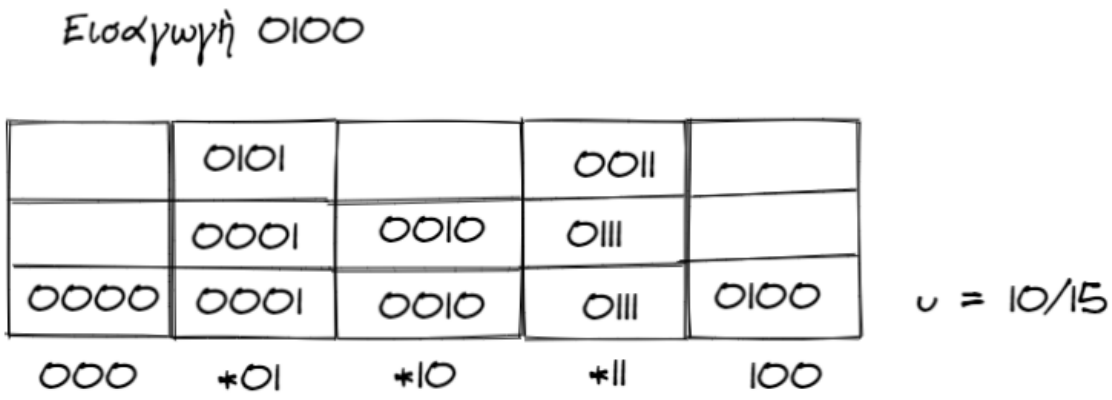
$\upsilon = 5/9$


Εισαγωγή 0010
01111
0101
0010
0001
0000
0001
υ = 6/9
0
1


Εισαγωγή 01111
01111
01111
0101
0010
0001
0000
0001
υ = 7/9
υ >= 70%
01111
01111
0101
0001
0000
0001
0010
υ = 7/12
0
1
00
*1
10


Εισαγωγή 0010
01111
01111
0101
0001
0010
0000
0001
0010
υ = 8/12
00
*1
10


Εισαγωγή 0011
0011
01111
01111
0101
0001
0010
0000
0001
0010
υ >= 70%
υ = 9/12
00
*1
10
0101
0001
0010
0011
0000
0001
0010
0011
υ = 9/12
00
01
10
11
υ >= 70%
0101
0001
0010
0011
0000
0001
0010
0011
υ = 9/15
000
*01
*10
*11
100



Άσκηση 6

a) Το ευρετήριο έχει ολικό βάθος 10, άρα 2^10 εγγραφές το μέγιστο, δλδ **1024 εγγραφές**.

b) Κάθε εγγραφή του ευρετηρίου είναι 4 bytes και έχουμε 1024 εγγραφές, άρα το μέγεθος του ευρετηρίου είναι **4096 bytes**.

c) Ένα bucket έχει χωρητικότητα 2400 bytes και χωράει 6 εγγραφές δεδομένων (2400 bytes/ 400 bytes). Έχουμε 1024 εγγραφές στο ευρετήριο τα οποία δείχνουν στα buckets. Ο μέγιστος αριθμός εγγραφών που μπορεί να περιέχει το αρχείο είναι:

(**2400 bytes / 400 bytes) * 1024 = 6144**

Άσκηση 7

Θα διαφωνήσω με το σκεπτικό του σχεδιαστή. Θέλουμε η συνάρτηση κατακερματισμού που θα εφαρμόσουμε να μοιράζει ομοιόμορφα τα δεδομένα στα buckets. Σε αυτήν την περίπτωση η επιλογή να χωρίσει με βάση τις τιμές του μισθού είναι λάθος. Είναι γνωστό πως οι περισσότεροι μισθοί κυμαίνονται σε ένα συγκεκριμένο εύρος π.χ [1000 … 2000]. Με αυτή την συνάρτηση θα έχουμε λίγα buckets που θα περιέχουν ένα μεγάλο μέρος των εγγραφών και πολλά buckets που θα είναι σχεδόν άδεια, άρα άνιση κατανομή των εγγραφών/ελλήνων εργαζόμενων. Ακόλουθο αυτού είναι να μην πετυχαίνεται η γρήγορη αναζήτηση που θα είχε σκοπό ο σχεδιαστής και να μην αξιοποιείται σωστά η δυναμική που προσφέρουν διάφορες τεχνικές κατακερματισμού.