

1η Σειρά Ασκήσεων, Σχεδιασμός Βάσεων Δεδομένων

p3170122 | Κων/νος Νικολούτσος

1΄ Άσκηση

Απο τα δεδομένα της εκφώνησης γνωρίζουμε ότι:

- Blocksize = 512 bytes
- Blocks_per_track = 20
- Tracks_per_surface = 400
- Rotate_speed = 2400rpm
- mean_seak_time = 30ms

α) Απάντηση:

Υπολογίζουμε τον χώρο που καταλαμβάνει μία εγγραφή:

$$size = 8 + 20 + 20 + 40 + 9 + 8 + 4 + 4 = 113 \text{ bytes}$$

Δεδομένου ότι πρέπει οι εγγραφες να αποθηκεύονται ολόκληρες στο block έχουμε ότι:

$$records = 512 / 113 \simeq 4.53 \Rightarrow records = 4$$

Άρα σε ένα block χωράνε 4 records.

β) Απάντηση:

Χρησιμοποιώντας τα αποτελέσματα του (α) έχουμε:

$$blocks = 20000 / 4 = 5000$$

Άρα χρειάζονται 5000 blocks για την αποθήκευση του αρχείου.

γ) Απάντηση:

i)

- Να πάει το HEAD στο σωστό track/cylinder
- Να περιμένουμε ένα μέσο rotational delay για να παει το HEAD στο σωστο block
- Να ξεκινάμε να διαβάζουμε τα μισά block εγγραφών (διότι κατα μέσο όρο τόσο θα χρειαστεί)

Υπολογίζω απο 2400rpm προκύπτει ότι σε 25ms κανει full rotation

$$\text{Άρα: } meanTime = 30ms + 12,5ms + 125tracks * 25ms = 3157,5ms$$

Όπου:

- 30ms το seektime
- 12.5 το average rotational delay
- 125 trucks για να διαβάσουμε τις μισες εγγραφες

ii)

Περιμένουμε να πάρει πολύ περισσότερο χρόνο γιατί θα κάνει πολλά random access.

$$meanTime = 2500blocks * (30ms + 12,5ms + 25ms/20records) = 109375ms$$

(*Υποθέτουμε ότι στο block αποθηκεύονται 4 εγγραφες)

δ) Απάντηση:

Γνωρίζουμε απο την ένταξή μας στην σχολή οτι η δυαδική αναζήτηση εχει average cost $\log(n)$ με βάση 2.

Αρα:

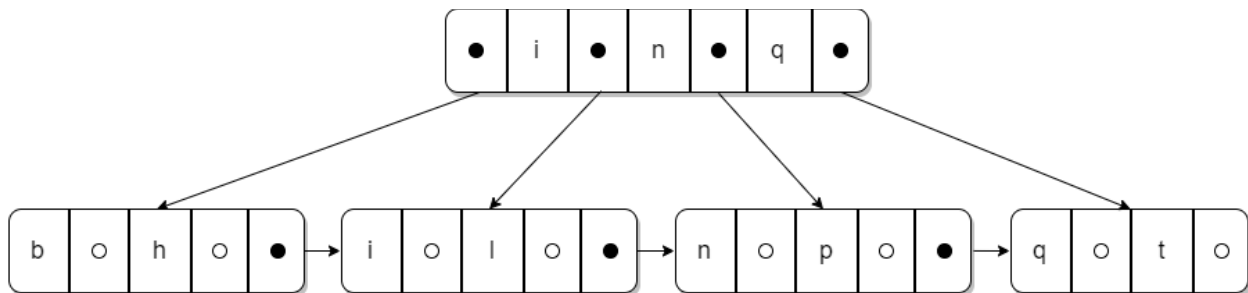
$$time = \log(2500) * (30ms + 12.5ms + 25/20) \simeq 493.5ms$$

* (Υποθέτουμε ότι στο block αποθηκεύονται 4 records)

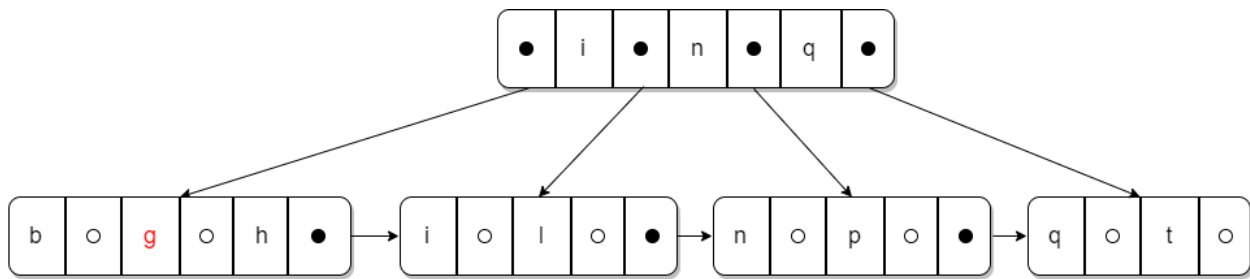
*(Το mean seek time το θεωρήσαμε 30ms εδω που σιγουρα στην πραξη θα ειναι καλυτερο ειδικα αν ειναι αποθηκευμενα συνεχόμενα στον δισκο και αυτο διοτι τα tracks θα βρισκονται κοντα στο αλλο καθώς συγκλίνει στο ζητούμενο record. Αρα η μετακίνηση της κεφαλής στα cylinders θα ειναι μικροτερη)

2' Ασκηση

α) Απάντηση:

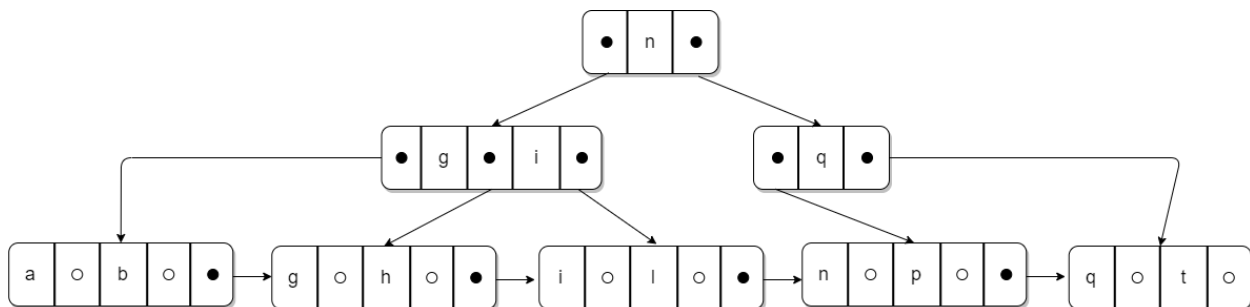


(Η αρχική κατάσταση του B+ tree)



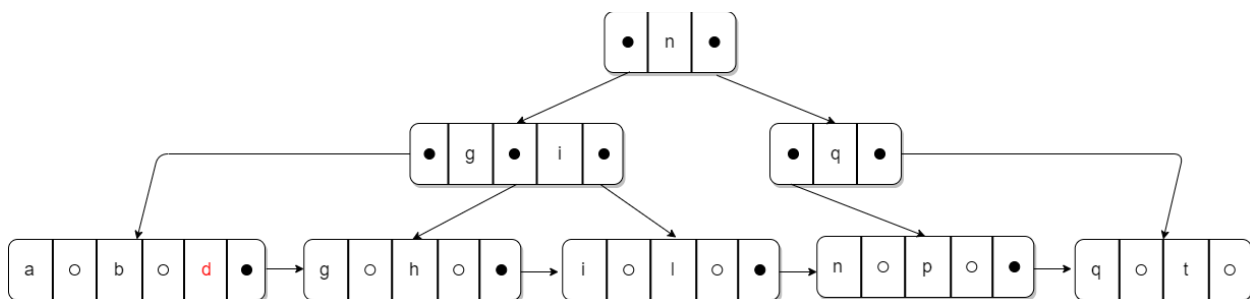
(Μετα την εισαγωγή του **G**)

Επεξήγηση: Το *g* είναι μικρότερο του *i* αρα ακολουθουμε το αριστερο reference και αφου το leaf εχει λιγοτερα απο 3 κλειδιά κάνουμε insert χωρις καποιο bottom-up σπάσιμο.



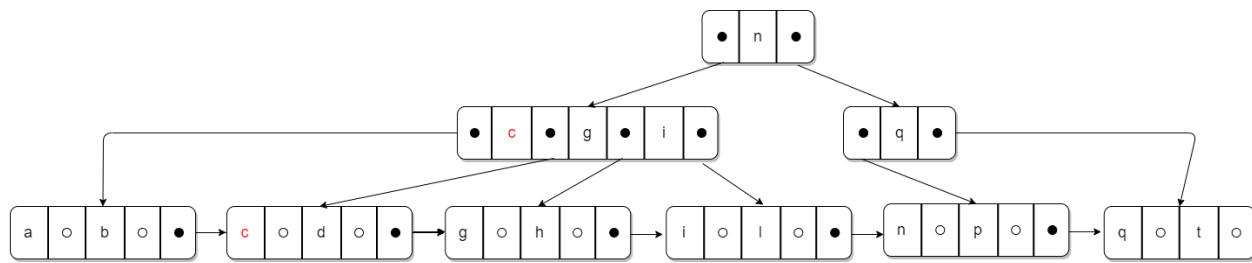
(Μετα την εισαγωγή του **A**)

Επεξήγηση: Το *a* είναι μικρότερο του *i* αρα ακολουθουμε το αριστερο reference και αφου το leaf εχει 3 κλειδιά κάνουμε σπασιμο και φτιάχνουμε έστρα leaf βάζοντας το *g,h* καθως βάζουμε στον parent node το *g*. Ομως χρειαζόμαστε άλλο ένα σπάσιμο διότι το root απέκτησε 3 κλειδιά. Ακολουθούμε την ίδια διαδικασία σπασίματος..



(Μετα την εισαγωγή του **D**)

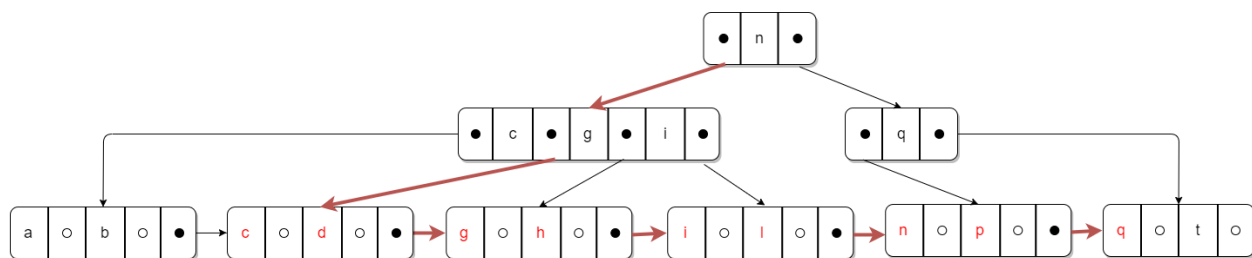
Επεξήγηση: Το *d* είναι μικρότερο του *n* αρα ακολουθουμε το αριστερο reference και ειναι μικρότερο του *g* αρα ακολουθούμε το αριστερό reference αφου το leaf εχει λιγοτερα απο 3 κλειδιά κάνουμε insert χωρις καποιο bottom-up σπάσιμο.



(Μετα την εισαγωγή του C)

Επεξήγηση: Το c είναι μικρότερο του n, άρα ακολουθούμε το αριστερό reference αφού το leaf έχει ήδη 3 κλειδιά κάνουμε το bottom-up σπασίμο που έχουμε περιγράψει παραπάνω.

β) Απάντηση:



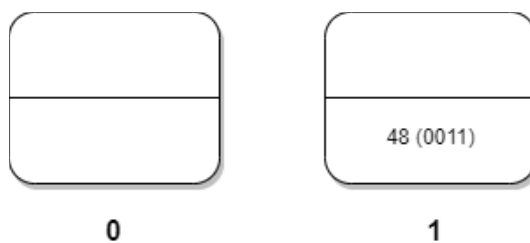
(Η παραπάνω εικόνα δείχνει αναλυτικά το μονοπάτι που θα ακολουθηθεί)

Άρα θα προσπελαστούν 7 κόμβοι.

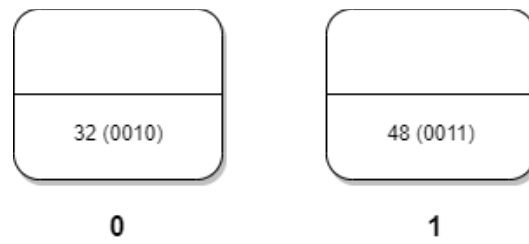
3^η Άσκηση

α) Απάντηση:

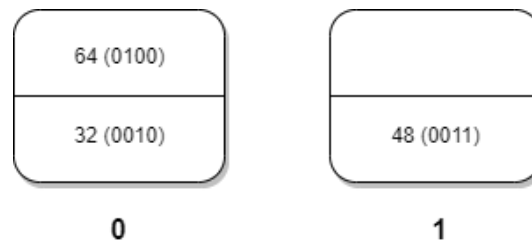
Ακολουθούν φωτογραφίες που εξηγούν αναλυτικά την διαδικασία του Linear hashing:



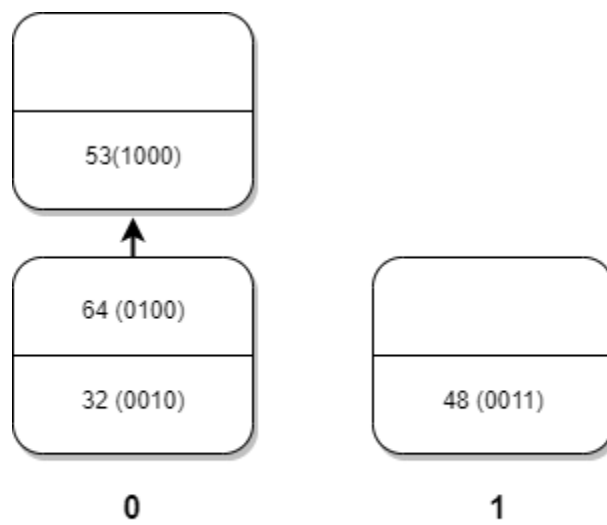
$$U = 1/4$$



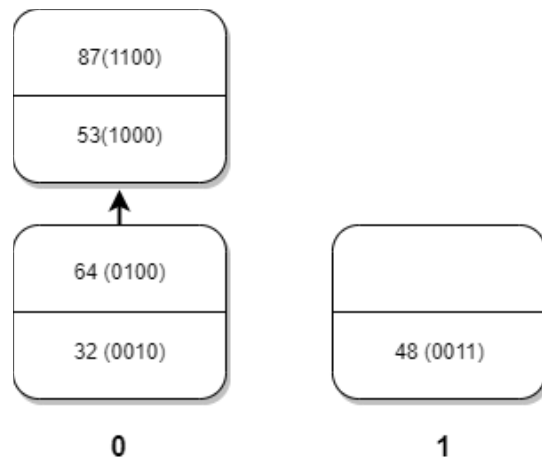
$$U = 2/4$$



$$U = 3/4$$

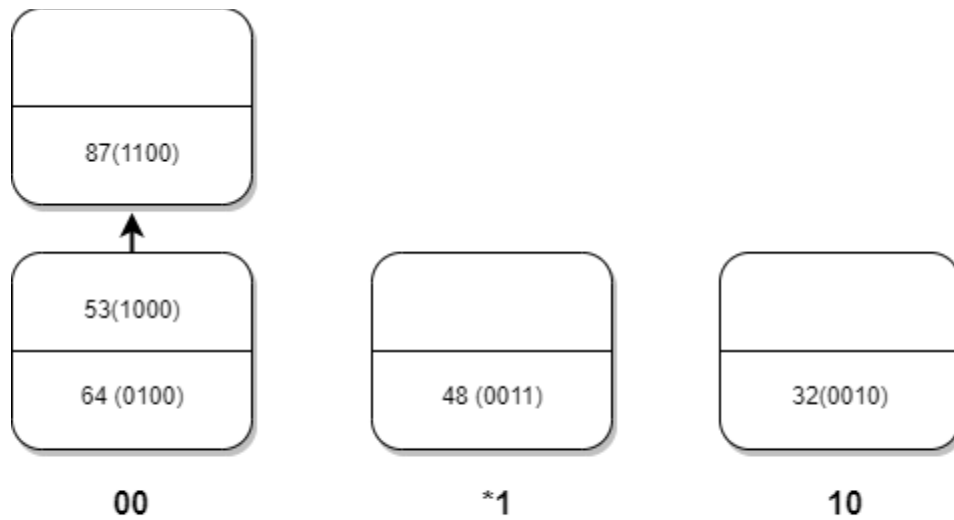


$$U = 4/6$$

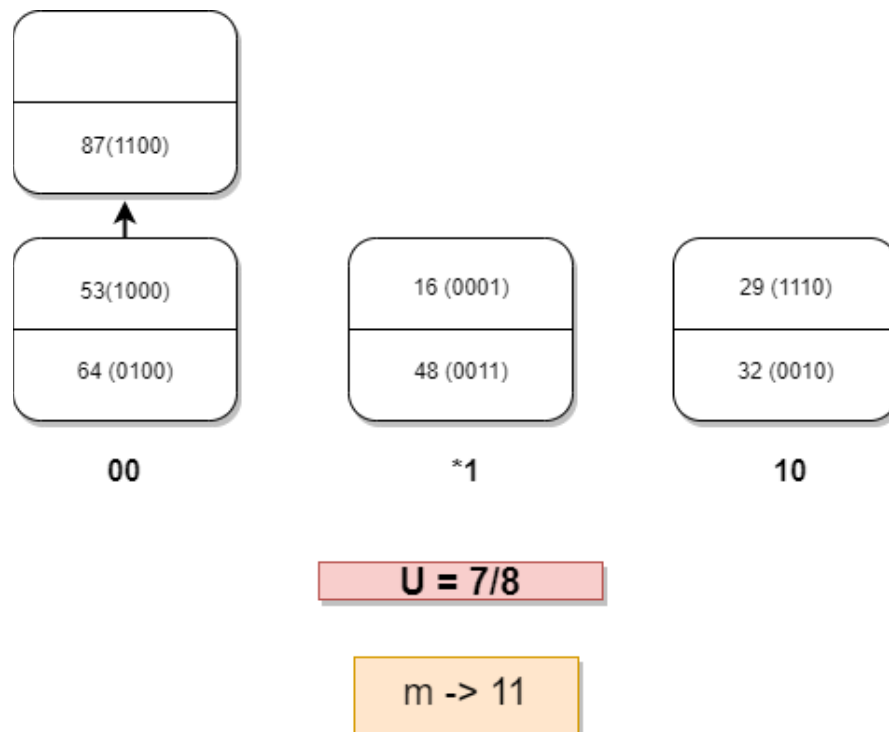
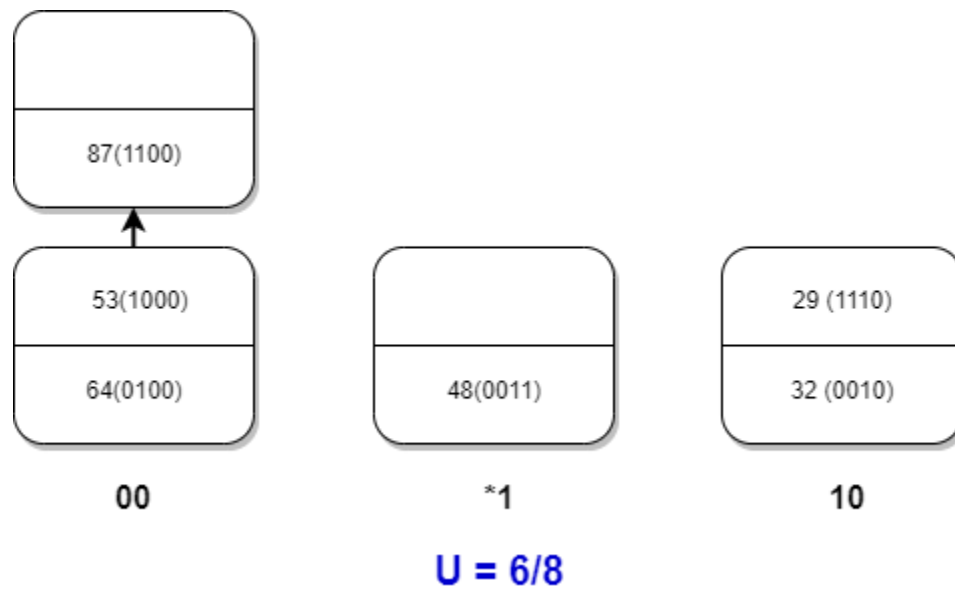


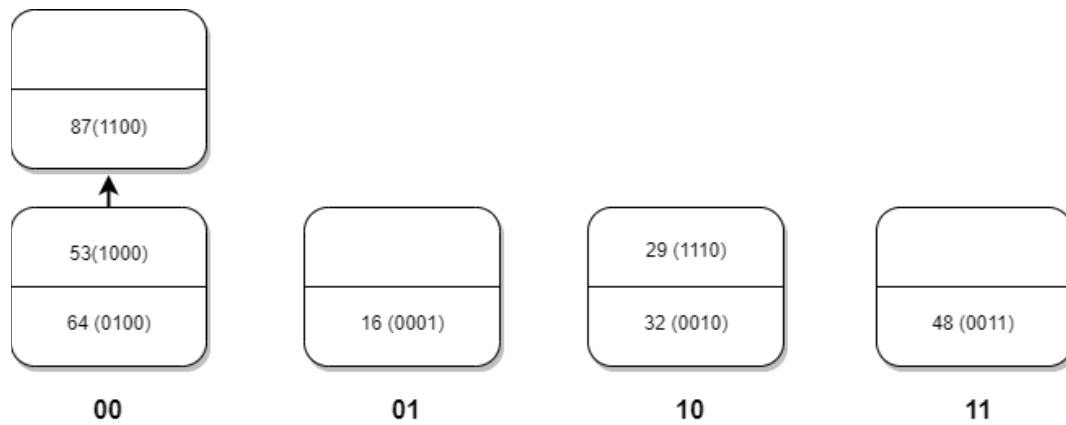
$$U = 5/6$$

m -> 10
i -> 2

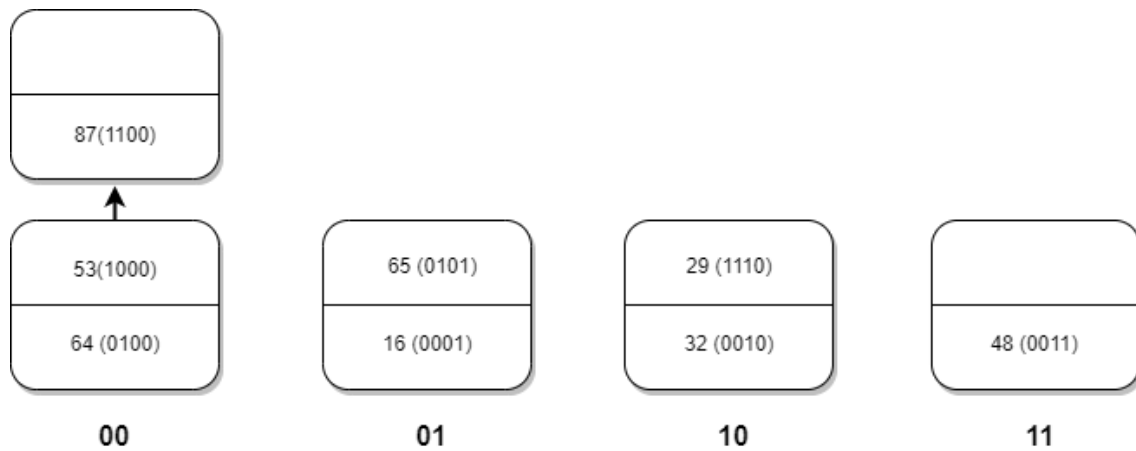


$$U = 5/8$$



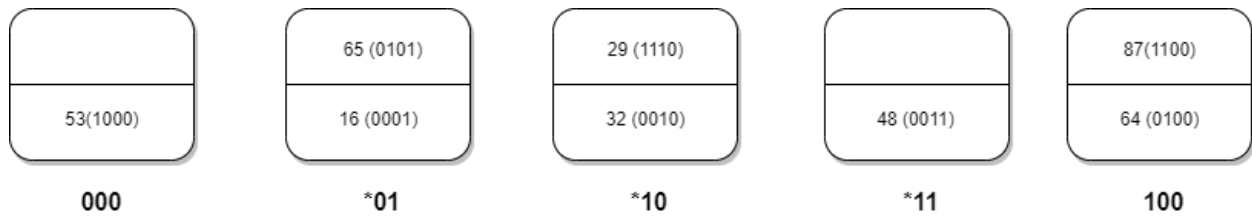


$$U = 7/10$$



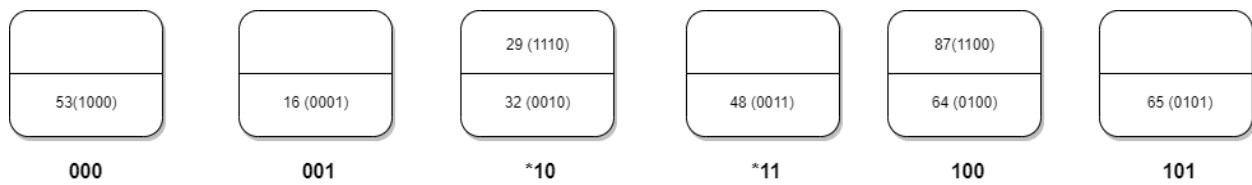
$$U = 8/10$$

m -> 100
i -> 3

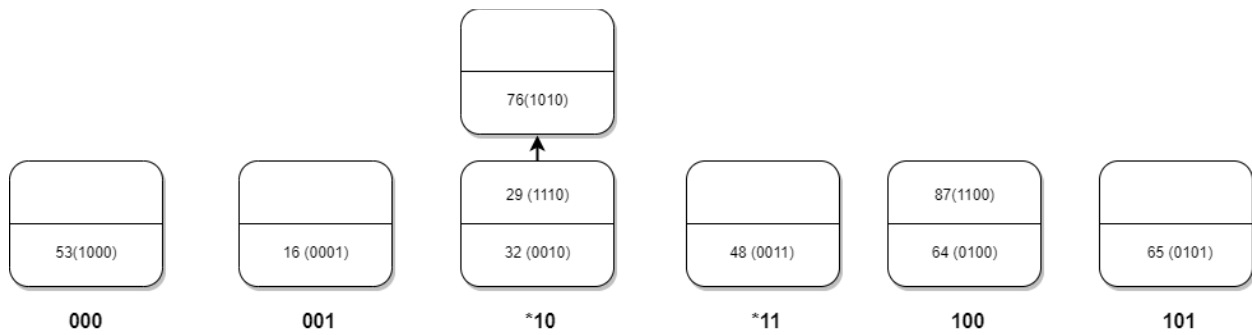


U = 8/10

m -> 101

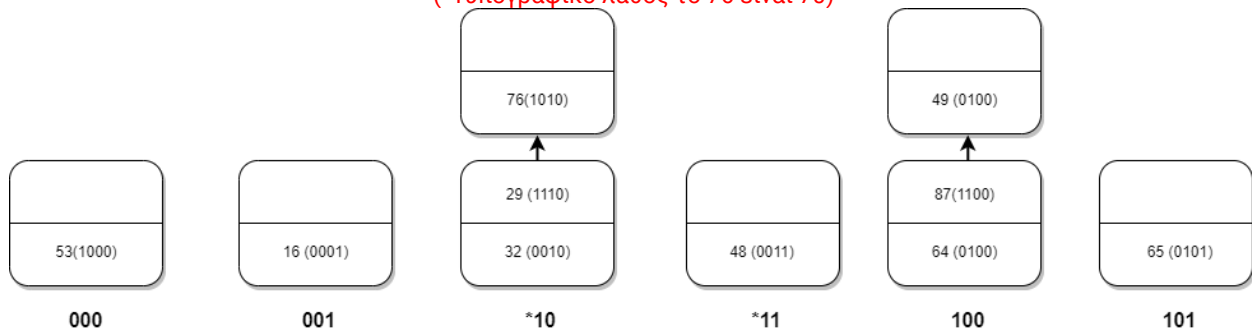


U = 8/12



U = 9/14

(*Τυπογραφικο λαθος το 76 ειναι 70)



U = 10/16

(*Τυπογραφικό λαθος το 76 είναι 70)

β) Απάντηση:

i) Για να υπολογίσουμε το ζητούμενο αρκεί να βρούμε τον μέσο όρο προσπελάσεων κάθε κλειδίου που υπάρχει στο linear hashing index.

$$\text{Αρα: } mean = (8*1 + 2*2) / 10 = 1,2$$

ii) Για να υπολογίσουμε αυτο αρκεί να κάνουμε την παρακάτω πράξη που εκφράζει τον μέσο όρο προσπελάσεων αν το κλειδί δεν ανήκει στο linear hashing index

$$mean = (1 + 1 + 2 + 1 + 2 + 1) / 6 = 8/6 = 1,33$$

4΄ Άσκηση bonus

Απάντηση:

(*Για να απαντήσουμε στο θέμα κρίνεται απαραίτητα να θεωρήσουμε ότι η hash function είναι αξιοπρεπή και χωρίζει τα data ομοιομορφα.)

Χρειάζονται περισσότερες από **1200 αναζητήσεις** για να συμφέρει η αναδιοργάνωση του αρχείου.

Στην παρακάτω εικόνα φαίνονται οι πράξεις όπου.

1. Ο χρόνος που παίρνουν N αναζητήσεις χωρίς αναδιοργάνωση. (Πολλαπλασιάζουμε με 3/2 διότι μπορεί στην αναζήτηση να μην την βρει στο 1ο page και να πρέπει να πάει στο 2ο.)
2. Ο χρόνος που θα κανεις για την αναδιοργάνωση του αρχείου και για να αναζητήσει αμέσως μετα N φορές. (Εδω δεν πολλαπλασιάζουμε με 3/2 γιατί είναι 1 page)
 - Με I συμβολίζουμε το transferTime

$$1. \text{ time} = N \cdot (\text{seekTime} + rd + \text{transferTime}) \cdot \frac{3}{2}$$

$$2. \text{ time}' = 600 \cdot (\text{seekTime} + rd + \text{χρονος μεταφοράς}) + N \cdot (\text{seekTime} + rd + \text{transferTime})$$

$$600(16 + 8,3 + 1) < \frac{1}{2}N(16 + 8,3 + 1) \Rightarrow$$

$$\Rightarrow N > \frac{1200 \cdot (16 + 8,3 + 1)}{16 + 8,3 + 1} \Rightarrow N > \underline{\underline{1200}}$$

Παρατήρηση ανεξάρτητα από ποσα RPM είναι το hard disk θα χρειαστούμε αναδιοργάνωση για περισσότερο από 1200 αναζητήσεις.