



Άσκηση 1

a) Χωρητικότητα σε bytes.

Για το ίχνος:

$$\frac{\text{bytes}}{\text{tracks}} = \frac{\text{bytes}}{\text{sector}} * \frac{\text{sectors}}{\text{tracks}} = 4096 * 256 = 1.048.320$$

Για την επιφάνεια:

$$\frac{\text{bytes}}{\text{surface}} = \frac{\text{bytes}}{\text{track}} * \frac{\text{tracks}}{\text{surface}} = 1.048.320 * 65536 = 68.702.699.520$$

Για τον δίσκο:

$$\frac{\text{bytes}}{\text{disk}} = \frac{\text{bytes}}{\text{surface}} * \frac{\text{surface}}{\text{disk}} = 68.702.699.520 * 8 * 2 = 1.099.243.192.320$$

Αφού κάθε πλακέτα έχει 2 επιφάνειες και ξέρουμε ότι οι πλακέτες είναι 8.

b) Ξέρουμε ότι ο αριθμός των κυλίνδρων είναι ίσος με τον αριθμό των tracks ανά surface, συνεπώς έχουμε 65536 κυλίνδρους.

c) Ξέρουμε πως η μέση καθυστέρηση περιστροφής είναι ίση με τον μισό χρόνο περιστροφής του δίσκου. Μάς δίνεται ότι η ταχύτητα περιστροφής είναι 7200 RPM. Συνεπώς,

Στα 60 seconds, γίνονται 7200 περιστροφές

Σε R seconds, γίνεται $\frac{1}{2}$ περιστροφή

$$R = \frac{30}{7200} = 4,17 \text{ msec}$$
 η μέση καθυστέρηση.

Για την μέγιστη καθυστέρηση, ισχύει

$$\frac{1}{7200} * 60 = 0,008 \text{ seconds}$$

d) Η χωρητικότητα του δίσκου σύμφωνα με το ερώτημα (a) είναι 1.048.320 bytes. Σε κάθε περιστροφή, διαβάζεται και μεταφέρεται στη RAM ένα ίχνος και η ταχύτητα περιστροφής είναι 7200 RPM. Συνεπώς ο ρυθμός μεταφοράς είναι

$$\frac{4096 * 256}{60/7200} = 125.829.120 \text{ bytes/seconds}$$

Άσκηση 2

Η σχέση R(#a,b,c,d,e) έχει μέγεθος

$$10+50+30+18+20 + 6 = 134 \text{ bytes.}$$

Άρα κάθε εγγραφή καταλαμβάνει 128 bytes. Ακόμα, ο pointer έχει μέγεθος 6 bytes και η σελίδα 1024 bytes.

a) Αν υποθέσουμε ότι μια εγγραφή δεν διασπάται κατά την αποθήκευση της σε παραπάνω από μία σελίδες και ότι μία σελίδα ισοδυναμεί με ένα block, έχουμε:

$$\frac{1024}{134} = 7$$

Συνεπώς, κάθε σελίδα χωράει 7 εγγραφές, αφήνοντας λίγο κενό και η σχέση R καταλαμβάνει $\lceil N/7 \rceil$ σελίδες ή blocks στον δίσκο.

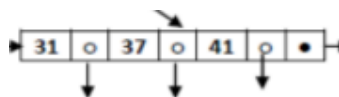
b) Στο sparse index, ο pointer δείχνει μόνο στο primary key, στην άσκηση μας στο #a. Επομένως, τα 16 bytes αποθηκεύονται σε σελίδες μεγέθους 1024 bytes. Οπότε,

$$\frac{1024}{16} = 64$$

Συνεπώς, κάθε σελίδα χωράει 64 εγγραφές, αφήνοντας λίγο κενό και η σχέση R καταλαμβάνει $\lceil N/64 \rceil$ σελίδες ή blocks στον δίσκο.

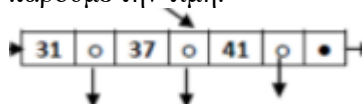
Άσκηση 3

1. Αναζήτηση της εγγραφής με κλειδί 41:
 - a. Το 41 είναι μεγαλύτερο του 13, συνεπώς πάμε δεξιά του 13
 - b. Το 41 είναι μεγαλύτερο του 31 αλλά μικρότερο του 43, άρα πάμε κάπου ενδιάμεσα
 - c. Βρίσκουμε το 41, το οποίο υπάρχει και ακολουθούμε την αναφορά για να πάρουμε την τιμή 41.



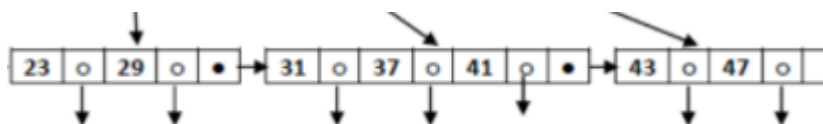
2. Αναζήτηση της εγγραφής με κλειδί 40:

- Το 40 είναι μεγαλύτερο του 13, πάμε δεξιά του 13
- Το 40 είναι μεγαλύτερο του 31 αλλά μικρότερο του 43, άρα πάμε κάπου ενδιάμεσα
- Το 40 δεν υπάρχει στο κελί που έπρεπε να είναι, οπότε δεν μπορούμε να πάρουμε την τιμή.



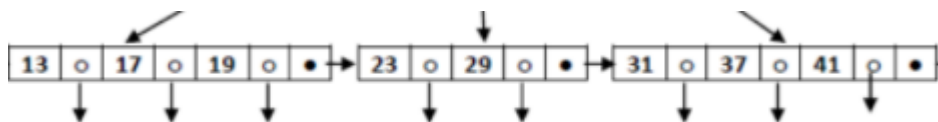
3. Αναζήτηση των εγγραφών με κλειδιά μικρότερα του 30.

- Το 30 είναι μεγαλύτερο του 13, συνεπώς πάμε δεξιά.
- Το 30 είναι μεγαλύτερο του 23 αλλά μικρότερο του 31, άρα πάμε κάπου ανάμεσα
- Στο τρίτο επίπεδο, διαβάζουμε αναγκαστικά όλα τα κλειδιά από το 23 και μετά, παρότι δεν είναι όλα μεγαλύτερα του 30.

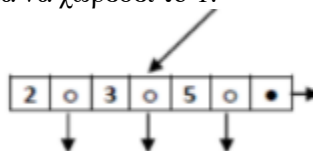


4. Αναζήτηση των εγγραφών με κλειδιά στο διάστημα [20,35].

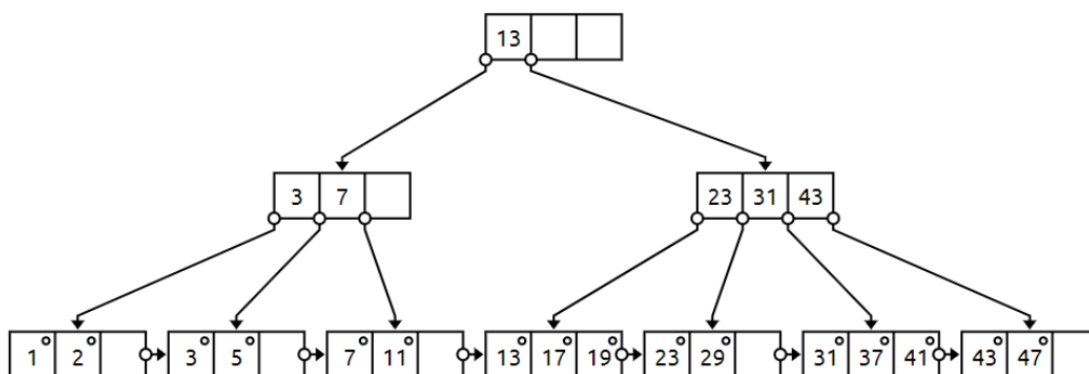
- Και το 20 και το 35 είναι μεγαλύτερα του 13, οπότε πάμε δεξιά.
- Το 20 είναι μικρότερο από το 23 και το 35 μικρότερο από το 43. Άρα πάμε ανάμεσα.
- Στο τρίτο επίπεδο συνεπώς, θα διαβάσουμε όλα τα κλειδιά με αρχή το κλειδί 13 μέχρι το κελί πριν το 43.



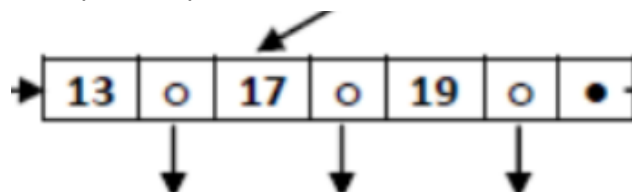
5. Εισαγωγή του κλειδιού με τιμή 1
- Το 1 είναι μικρότερο του 13, οπότε πάμε αριστερά.
 - Το 1 είναι μικρότερο του 7, οπότε πάμε αριστερά.
 - Στο τρίτο επίπεδο, η σελίδα είναι γεμάτη, οπότε πρέπει να την τροποποιήσουμε για να χωρέσει το 1.



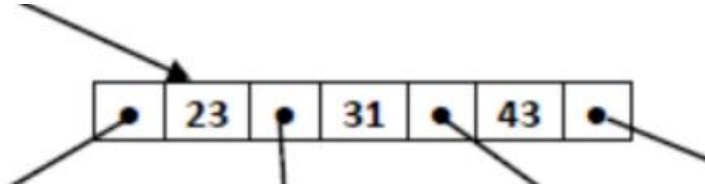
- Σπάμε την τριάδα, ανεβαίνει το 3 στο 2^ο επίπεδο μαζί με το 7 και γίνεται 1,2,5
- Το υπόλοιπο δέντρο δεν αλλάζει.



6. Διαδοχική εισαγωγή τριών κλειδιών με τιμές 14,15 και 16 αντίστοιχα.
- Και τα τρία είναι μεγαλύτερα του 13, οπότε θα πάμε δεξιά.
 - Και τα τρία είναι πιο μικρά από το 19, οπότε θα πάμε αριστερά.
 - Η σελίδα είναι γεμάτη, οπότε δεν μπορούν να προστεθούν τα 14,15,16 συνεπώς πρέπει να τροποποιηθεί.



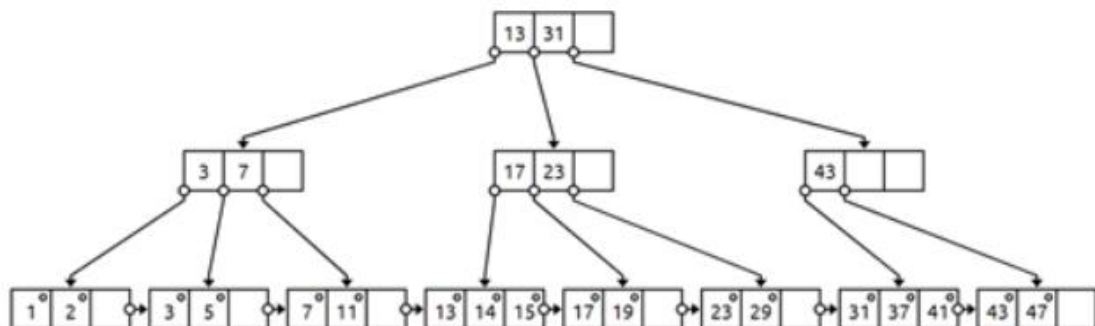
- d. Σπάμε την τριάδα και το 17 ανεβαίνει πάνω, μαζί με τα 23,31,43. Ωστόσο αυτό δεν μπορεί να συμβεί καθώς και η προηγούμενη σελίδα είναι γεμάτη. Οπότε πρέπει και αυτή να σπάσει και να ανέβει το 31 στην ρίζα.



- e. Με αυτή την διαδικασία, προστίθεται το 14.

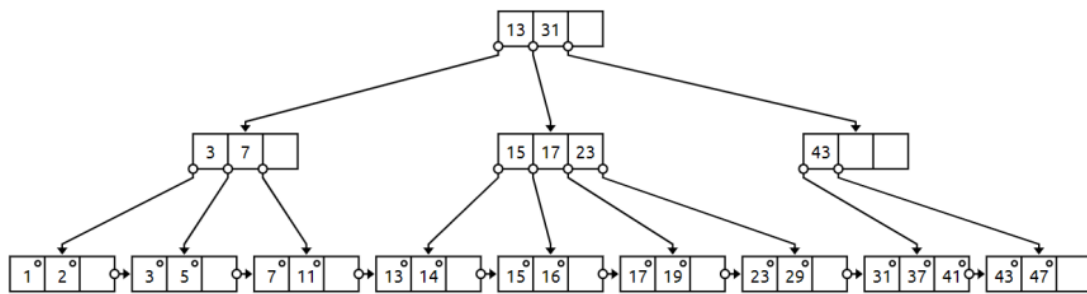


- f. Στη συνέχεια προσθέτουμε και τα 15, χωρίς να αλλάζει το σχήμα αφού θεωρούμε ότι $n=3$ και υπάρχει ελεύθερη θέση στο συγκεκριμένο block του δέντρου.



- g. Τέλος προσθέτουμε και το 16. Διασπάται η τριάδα 13,14,15 και ανεβαίνει το 15 πάνω.

Το δέντρο μετά από όλα τα βήματα:



Για τα B+ trees χρησιμοποιήθηκε το site <https://projects.calebevans.me/b-sketcher/>

Άσκηση 4

Από τα δεδομένα της άσκησης ξέρουμε ότι το μέγεθος της σελίδας είναι 2048 bytes, ο μέγεθος του δείκτη προς τον κόμβο του δέντρου είναι 12 bytes, ο δείκτης προς την εγγραφή 12 bytes και το πρωτεύον κλειδί 8 bytes. Συνεπώς, το άθροισμα από την ρίζα μέχρι την εγγραφή είναι $12+12+8 = 32$ bytes

Κάθε κόμβος του B+ tree περιέχει το πολύ $n+1$ pointers και n keys. Οπότε ισχύει:

$$8 * (n) + 12 * (n + 1) \leq 2048$$

$$n \leq 102$$

Ακόμα, η ρίζα έχει $n-1$ pointers. Άρα, ο μέγιστος αριθμός των εγγραφών της σχέσης R που μπορούν να ευρετηριαστούν με ένα ευρετήριο B+ δέντρου τριών επιπέδων είναι

$$101 * 102 * 102 = 1.050.804 \text{ εγγραφές}$$

Άσκηση 5

Έχουμε 2 κάδους ($m=1$) χωρητικότητας 3 εγγράφων έκαστος. Για την κατανομή τους χρησιμοποιούμε το τελευταίο από δεξιά bit.. Όταν $U \geq 70\%$, αυξάνουμε τον αριθμό των κάδων. Αρχικά το $i \rightarrow 1$.

1. Εισαγωγή 0000

$$\text{Utilization} = 1/6 \leq 70\%$$

0000	
0	1

2. Εισαγωγή 0001

Utilization = $2/6 \leq 70\%$

0000	0001
0	1

3. Εισαγωγή 0001

Utilization = $3/6 \leq 70\%$

	0001
0000	0001
0	1

4. Εισαγωγή 0101

Utilization = $4/6 \leq 70\%$

	0101
	0001
0000	0001
0	1

5. Εισαγωγή 0111

Utilization = $5/6 > 70\%$

Εφόσον ξεπεράσαμε το όριο του Utilization, πρέπει να μεγαλώσουμε το αρχείο. Άρα,
m->11, i ->2, U=5/9 $\leq 70\%$

	0101	
	0001	
0000	0001	0111
*0	01	11

6. Εισαγωγή 0010

Utilization = $6/9 \leq 70\%$

	0101	
0010	0001	
0000	0001	0111
*0	01	11

7. Εισαγωγή 0111

$$\text{Utilization} = 7/9 > 70\%$$

Εφόσον ξεπεράσαμε το όριο του Utilization, πρέπει να μεγαλώσουμε το αρχείο. Άρα,

$$m \rightarrow 10, i \rightarrow 2, U = 7/12 \leq 70\%$$

	0101		
	0001	0010	0111
0000	0001		0111
00	01	10	11

8. Εισαγωγή 0110

$$\text{Utilization} = 8/12 \leq 70\%$$

	0101		
	0001	0010	0111
0000	0001	0110	0111
00	01	10	11

9. Εισαγωγή 0011

$$\text{Utilization} = 9/12 > 70\%$$

Εφόσον ξεπεράσαμε το όριο του Utilization, πρέπει να μεγαλώσουμε το αρχείο. Άρα,

$$m \rightarrow 100, i \rightarrow 3, U = 9/15 \leq 70\%$$

	0101		0011	
	0001	0010	0111	
0000	0001	0110	0111	
000	*01	*10	*11	100

10. Εισαγωγή 0100

$$\text{Utilization} = 10/15 \leq 70\%$$

	0101		0111	
	0001	0010	0111	
0000	0001	0110	0011	0100
000	*01	*10	*11	100

Άσκηση 6

- a) Το βάθος του ευρετηρίου είναι 10, συνεπώς ο μέγιστος αριθμός εγγραφών στο ευρετήριο είναι $2^{10} = 1024$. Το παραπάνω ισχύει κάνοντας την παραδοχή ότι δεν είναι αναγκαίο κάθε εγγραφή να δείχνει και σε διαφορετικό κάδο. Αυτό εξαρτάται και από το τοπικό βάθος των κάδων.
- b) Αφού το βάθος του ευρετηρίου είναι 10, το ευρετήριο έχει μέγεθος $2^{10} = 1024$ νούμερα και κάθε εγγραφή του ευρετηρίου έχει μέγεθος 4 bytes. Οπότε το μέγεθος του ευρετηρίου είναι $1024 * 4 = 4096$ bytes.
- c) Ο κάδος έχει μέγεθος 2400 bytes και κάθε εγγραφή δεδομένων 400 bytes, οπότε μέχρι 6 εγγραφές μπορεί να περιέχει ο κάδος. Έχουμε $2^{10} = 1024$ κάδους, οπότε το αρχείο δεδομένων μπορεί να περιέχει $1024 * 6 = 6144$ εγγραφές.

Άσκηση 7

Η συνάρτηση κατακερματισμού που έχει επιλέξει ο σχεδιαστής του υπουργείου εργασίας είναι μη αποτελεσματική στην συγκεκριμένη περίπτωση. Δεν μπορούμε να πετύχουμε ομοιόμορφη κατανομή των εγγραφών στους κάδους με το σκεπτικό ότι προσθέτουμε σε κάθε κάδο τον ίδιο αριθμό κλειδιών, καθώς οι μισθοί δεν είναι ίσα κατανεμημένοι στον πληθυσμό. Αυτό απορρέει από το γεγονός ότι δεν παίρνει ίσο ποσοστό ανθρώπων τα ίδια χρήματα. Για παράδειγμα, οι άνθρωποι που ο μισθός τους είναι μικρότερος από 2000€, είναι πολύ περισσότεροι από εκείνους που έχουν μισθό μεγαλύτερο από 4000€. Συνεπώς, ακολουθώντας την προτεινόμενη συνάρτηση κατακερματισμού, θα γεμίζουν πολύ γρήγορα οι κάδοι που περιέχουν μικρότερους μισθούς, ενώ οι κάδοι με μεγαλύτερους μισθούς θα είναι πιο άδειοι. Αποτέλεσμα αυτού είναι η μείωση της επίδοσης για τους πρώτους και η σπατάλη μνήμης για τους δεύτερους. Συνεπώς, δεν συμφωνούμε με το σκεπτικό του σχεδιαστή.