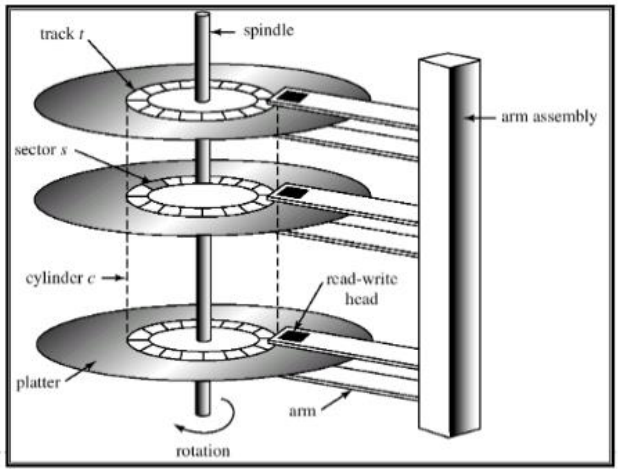
1η Σειρά Ασκήσεων

ΚΑΡΑΓΕΩΡΓΗΣ ΠΛΑΤΩΝ 3180068

1Η Άσκηση

Α) 1 ίχνος έχει 256 τομείς και το μέγεθος ενός τομέα είναι 4096 bytes άρα η χωρητικότητα ενός ίχνους σε bytes είναι 256*4096 = 1,048,576. Αντίστοιχα, αυτόν τον αριθμό τον πολλαπλασιάζουμε με 65,536 γιατί τόσα ίχνη έχουμε ανά επιφάνεια. Άρα η χωρητικότητα της επιφάνειας σε bytes είναι 65,536*1,048,576 = 68,719,476,736. Τέλος, έχουμε 8 πλακέτες διπλής όψης άρα 16 επί τον αριθμό από πριν για να βρούμε τη χωρητικότητα του δίσκου σε bytes. Συγκεκριμένα, 16* 68,719,476,736 = 1,099,511,627,776.

Β) Ο αριθμός των κυλίνδρων ισούται με τον αριθμό των ιχνών. Το καταλαβαίνουμε και από το παρακάτω σχήμα:



Άρα έχουμε 65,536 ίχνη ανά επιφάνεια και 16 επιφάνειες επομένως έχουμε 1,048,576 ίχνη και άρα κυλίνδρους.

Γ) Από τη θεωρία, χρειάζομαι να περιμένω μισή περιστροφή κατά μέσο όρο. Άρα αν βρώ το χρόνο που χρειάζεται η μισή περιστροφή είμαι καλυμμένος. Αυτό το βρίσκω από τον τύπο 60*0.5/7200= 4,17msecs. Αυτή είναι και η μέση καθυστέρηση περιστροφής (rotational delay). Η μέγιστη καθυστέρηση περιστροφής θα είναι όταν χρειαστεί να γίνει μία πλήρης περιστροφή. Άρα το διπλάσιο από πριν, δηλαδή 4,17*2= 8,34msecs.

Δ) Για να βρω το rate σκέφτηκα ως εξής. Σε μία περιστροφή μεταφέρεται ένα ίχνος και η χωρητικότητα αυτού από το 1α είναι 1,048,576 bytes. Άρα από τη φυσική, χρειαζόμαστε το χρόνο που χρειάζεται μία περιστροφή για να βρούμε το ρυθμό μεταφοράς. Σε 1 δευτερόλεπτο

κάνει 7200/60= 120 περιστροφές. Σε Χ δευτερόλεπτα κάνει 1 περιστροφή. Άρα σε 1/120 seconds γίνεται μία περιστροφή. Επομένως έχουμε 1,048,576/(1/120) = 125,829,120 bytes/second.

2Η Άσκηση

Α) Έχω 134 bytes ανά γραμμή γιατί είναι 50+30+18+20+6 bytes από τα γνωρίσματα και από τον δείκτη. Το block είναι ίσο με τη σελίδα επομένως έχει μέγεθος 1024 bytes. Άρα έχουμε 1024/134=7.6 και άρα κάθε block χωράει 7 ολόκληρες εγγραφές. Έχουμε Ν εγγραφές συνολικά άρα η σχέση R και πυκνού ευρετηρίου είναι Ν/7.

Β) Στη μέθοδο sparse έχουμε 1 pointer ανά data block άρα χρειαζόμαστε 6 bytes για pointer. Ωστόσο χρειαζόμαστε και το Primary Key που έχει 10 bytes μέγεθος. Άρα έχουμε 10 + 6 = 16 απαραίτητα bytes και συνεπώς έχουμε 1024/16 = 64 εγγραφές ανά block. Άρα η σχέση R και αραιού ευρετηρίου είναι Ν/64.

3Η Άσκηση

1) Αναζητώ την εγγραφή με τιμή 41 και συνεπώς όταν ξεκινήσω την αναζήτηση από τη ρίζα με τιμή 13, θα κατευθυνθώ προς τα δεξιά διότι το 41 είναι προφανώς μεγαλύτερο. Στα δεξιά βρίσκουμε έναν 3-κόμβο και έχουμε 4 πιθανές κατευθύνσεις. Το 41 είναι μεγαλύτερο του 23 αλλά και του 31, δεν είναι όμως του 43 επομένως κατευθυνόμαστε στον 3-κόμβο-φύλλο και εκεί εντοπίζουμε το 41.

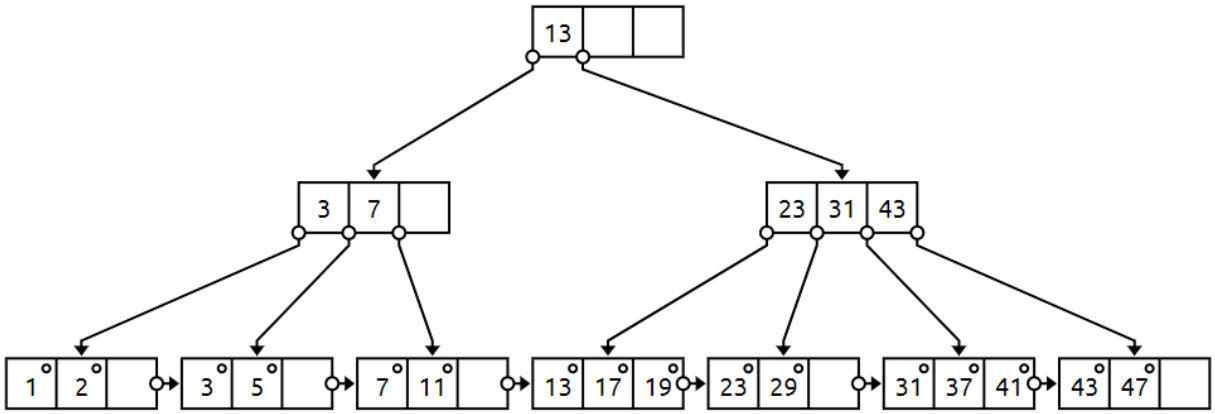
2) Αναζητώ την εγγραφή με τιμή 41 και συνεπώς όταν ξεκινήσω την αναζήτηση από τη ρίζα με τιμή 13, θα κατευθυνθώ προς τα δεξιά διότι το 41 είναι προφανώς μεγαλύτερο. Στα δεξιά βρίσκουμε έναν 3-κόμβο και έχουμε 4 πιθανές κατευθύνσεις. Το 41 είναι μεγαλύτερο του 23 αλλά και του 31, δεν είναι όμως του 43 επομένως κατευθυνόμαστε στον 3-κόμβο-φύλλο και εκεί ξεκινάμε από το 31 και φτάνουμε στο 41 χωρίς να βρούμε το 40 άρα δεν υπάρχει στις εγγραφές.

3) Αναζητώ τις εγγραφές με κλειδιά μικρότερα του 30. Αρχικά ξεκινώ από τη ρίζα και πηγαίνω αριστερά στον κόμβο 7 και στους κόμβους φύλλα 2-3-5 και 7-11. Έπειτα πηγαίνω στον δεξί κόμβο από τη ρίζα καθώς το 30 είναι μεγαλύτερο του 13. Στον 3-κόμβο από κάτω βλέπω τις τιμές 23 και 31 και διαλέγω να διαλέξω την ενδιάμεση κατεύθυνση τους καθώς το 30 είναι ανάμεσα σε αυτές τις 2 τιμές. Εκεί παρατηρώ ότι η μεγαλύτερη τιμή του 2-κόμβου φύλλο είναι 29. Συνεπώς, βρήκαμε το άνω άκρο των τιμών που πρέπει να επιστρέψουμε και επιστρέφουμε όλες τις τιμές που βρήκαμε μέχρι και το 29.

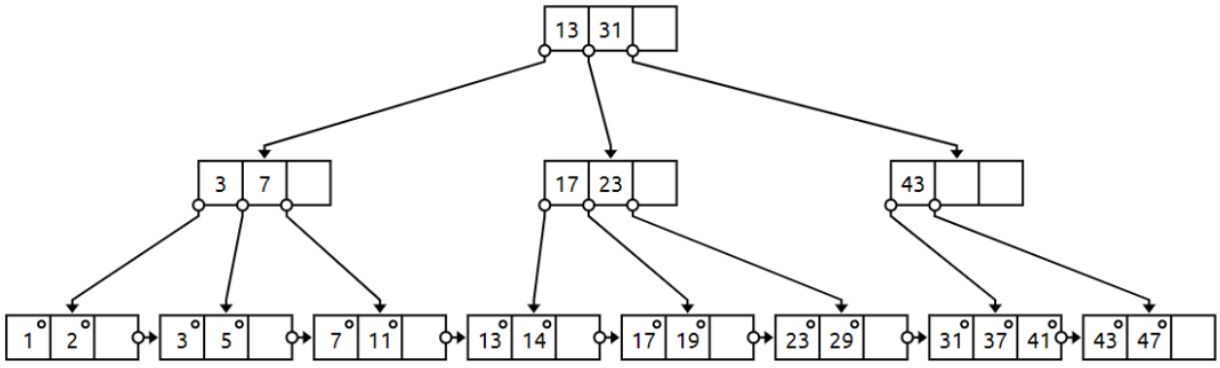
4) Αρχικά πηγαίνω δεξιά του 13 καθώς και το 20 και το 35 είναι μεγαλύτερα από αυτό. Έπειτα πηγαίνουμε στον 3-κόμβο από κάτω και βρίσκουμε το 23 που είναι μεγαλύτερο του 20 άρα συνεχίζουμε στα φύλλα και εντοπίζουμε τον 3-κόμβο 13-17-19. Άρα δεν έχουμε τιμή μεταξύ του 20 και του 23 και επομένως επιστρέφουμε στον κόμβο-3 του ενδιάμεσου επιπέδου.

Παρατηρούμε ότι μεταξύ 23 και 30 υπάρχουν τιμές που είναι εντός του πεδίου ορισμού μας επομένως κατεβαίνουμε στο τελευταίο επίπεδο, συλλέγουμε τις τιμές και ξαναεπιστρέφουμε στον ενδιάμεσο κόμβο. Εκεί προχωράμε στην κατεύθυνση μεταξύ των τιμών 31 και 43 και στο επίπεδο των φύλλων κρατάμε μόνο το 31 καθώς μετά έχει το 37 που είναι εκτός του πεδίου ορισμού. Η αναζήτηση μας τελείωσε.

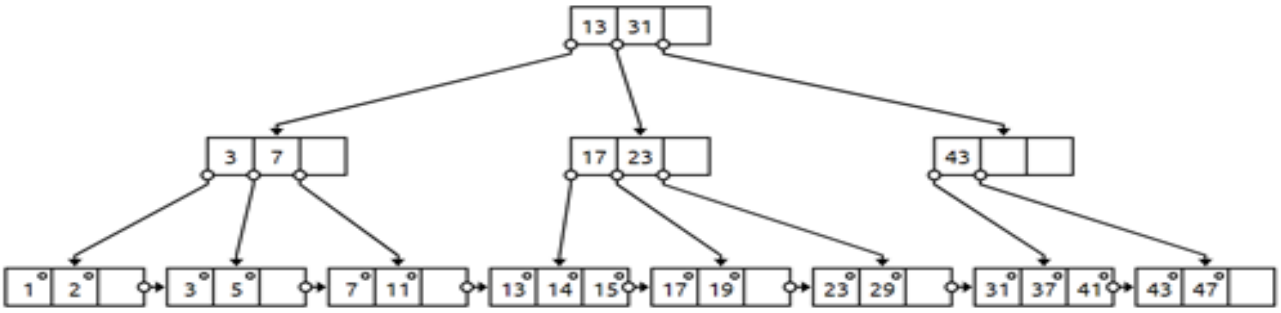
5) Θέλω να προσθέσω το κλειδί 1. Πηγαίνω από τον κόμβο 13 αριστερά στον κόμβο 7. Έπειτα από τον κόμβο 7 πηγαίνω αριστερά και βρίσκω ότι η μικρότερη τιμή που έχω είναι το 2. Κατ'επέκταση, προσθέτω το 1 αριστερά από το 2 αφού είναι μικρότερο, όμως επειδή είναι 3-κόμβος ήδη (2-3-5) διασπάται και το 5 πηγαίνει πάνω. Το αποτέλεσμα αυτής της διαδικασίας φαίνεται στο παρακάτω σχήμα:



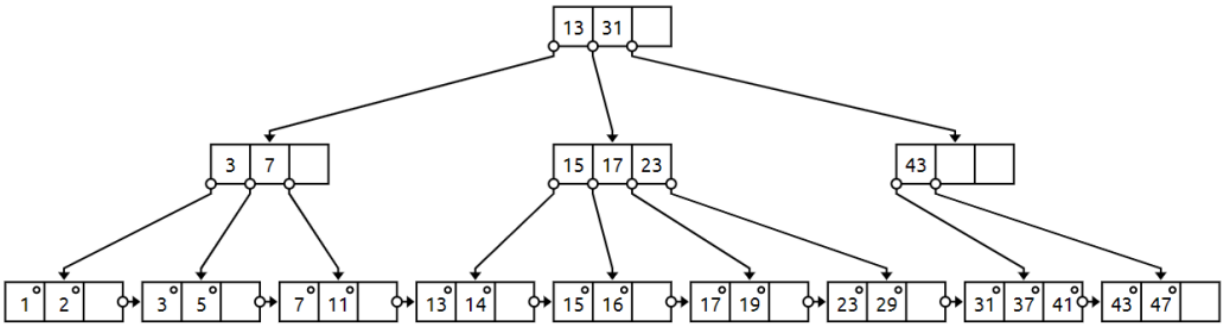
6) Θέλω να προσθέσω το κλειδί 14. Κατευθύνομαι στον 3-κόμβο 13-17-19 και βλέπω ότι πρέπει να προσθέσω το 14 εκεί. Πάλι ο κόμβος πρέπει να διασπαστεί επομένως το 17 πηγαίνει πάνω και τα 13-14-19 περιμένουν να γίνουν οι πάνω ανακατατάξεις για να κατανεμηθούν κατάλληλα. Τώρα πρέπει να διασπαστεί και αυτός ο κόμβος γιατί είναι και αυτός ήδη 3-κόμβος (23-31-43). Άρα το 31 πηγαίνει πάνω και η ρίζα πλέον είναι ο 2-κόμβος 13-31 και έχει 3 παιδιά, το 3-7 για αριστερό, το 17-23 για μεσαίο και το 43 για δεξί. Έτσι, ο κόμβος 13-14 θα γίνει αριστερό παιδί του κόμβου 17-23 και το 19 θα προσθεθεί στον μεσαίο κόμβο (17-19). Το αποτέλεσμα φαίνεται στο σχήμα παρακάτω.



Ακολουθεί η προσθήκη του κλειδιού 15. Πλέον είναι απλή διαδικασία, το προσθέτω στον 2-κόμβο 13-14 και γίνεται 3-κόμβος 13-14-15. Το αποτέλεσμα φαίνεται παρακάτω:



Τέλος, έχουμε την προσθήκη του 16 που πρέπει να προστεθεί στον 3-κόμβο 13-14-15 επομένως διασπάται και το 15 πηγαίνει πάνω στον 2-κόμβο 17-23 ο οποίος τώρα θα γίνει 3-κόμβος. Το 13-14 θα είναι το αριστερό παιδί, το 15 16 θα είναι το μέσα αριστερά, το 17 19 θα είναι το μέσα δεξιά και το δεξί θα είναι το 23 29. Το βλέπουμε και στο σχήμα παρακάτω:



Τα σχήματα δημιουργήθηκαν στο σάιτ:

https://projects.calebevans.me/b-sketcher/

4η 'Ασκηση

Σε ένα B+ δέντρο έχουμε από τις διαφάνειες n κόμβους και n+1 pointers. Άρα έχουμε 8n + 12(n+1)<= 2048 αφού από την εκφώνηση κοστίζουν και τα δύο είδη pointer 12 bytes και κάθε κλειδί 8 bytes και αυτά πρέπει να χωράνε σε μία σελίδα που έχει 2 KB μέγεθος. Επομένως, προκύπτει n<=101,8 δηλαδή nmax = 101. Από εκεί ξέρουμε ότι η ρίζα έχει n πιθανούς pointer, το ενδιάμεσο στάδιο του δέντρου έχει n*(n+1) pointers και τα φύλλα έχουν n*(n+1)*(n+1) pointers. Άρα συνολικά ο μέγιστος αριθμός εγγραφών αντιστοιχεί στον μέγιστο πιθανό αριθμό pointers, δηλαδή n*n*(n-1) με n = 101. Δηλαδή ο μέγιστος αριθμός εγγραφών είναι 1,050,804.

5Η Άσκηση

Σε αυτή την άσκηση θα κάνω ένα σχήμα σε κάθε βήμα και παράλληλα αν χρειάζεται θα εξηγώ.

*1Η ΕΙΣΑΓΩΓΗ -> **0000***

0000	
0	**1**

*2^{H} ΕΙΣΑΓΩΓΗ -> **0001***

0000	0001
0	**1**

*3^{H} ΕΙΣΑΓΩΓΗ -> **0001***

	0001
0000	0001
0	**1**

*4^{H} ΕΙΣΑΓΩΓΗ -> **0101***

	0101
	0001
0000	0001
0	**1**

Εδώ έχουμε 66% χωρητικότητα και θέλουμε να προσθέσουμε στον δεξί πίνακα αλλά δεν έχουμε χώρο. Θα χρησιμοποιήσουμε μία σελίδα υπερχείλισης.

*5^{H} ΕΙΣΑΓΩΓΗ -> **0111***

0111

ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ BIT=1

	0101
	0001
0000	0001
0	**1**

Έχουμε πληρότητα σε ποσοστό 5/9 άρα μπορούμε να προχωρήσουμε.

*6Η ΕΙΣΑΓΩΓΗ -> **0010***

0111

ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ BIT=1

	0101
0010	0001
0000	0001
0	**1**

*7Η ΕΙΣΑΓΩΓΗ -> **0111***

0111
0111

ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ BIT=1

	0101
0010	0001
0000	0001
0	**1**

Πλέον έχουμε πληρότητα άνω του ορίου επομένως πρέπει να αυξήσουμε το m. Θα γίνει από 1 σε 10 και το i θα γίνει 2.

0111
0111

*ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ BIT=*1*

	0101
	0001
0000	0001

00 ***1**

0010

10

*8Η ΕΙΣΑΓΩΓΗ -> **0010***

Εδώ πλέον έχουμε προετοιμάσει το έδαφος για την επόμενη προσθήκη και γίνεται εύκολα.

0111
0111

*ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ BIT=*1*

	0101
	0001
0000	0001

00 ***1**

0010
0010

10

*9Η ΕΙΣΑΓΩΓΗ -> **0011***

0011
0111
0111

*ΣΕΛΙΔΑ ΥΠΕΡΧΕΙΛΙΣΗΣ ΓΙΑ ΒΙΤ=*1*

	0101
	0001
0000	0001

00 ***1**

0010
0010

10

Τώρα έχουμε χωρητικότητα 9/12 και άρα πρέπει να γίνει το m από 10 σε 11. Έτσι, δεν χρειαζόμαστε πλέον την σελίδα υπερχείλισης. Αναμένουμε όμως πάλι να έχουμε υπερχείλιση γιατί θα έχουμε ακόμα 12 θέσεις και 9 στοιχεία, επομένως το m γίνεται από 11 σε 100 και πλέον και το i γίνεται 3.

	0101
	0001
0000	0001

000 ***01**

	0011
0010	0111
0010	0111

***10** ***11**

100

*10^{H} ΕΙΣΑΓΩΓΗ -> **0100***

Έχοντας προετοιμαστεί για την επόμενη εισαγωγή από πριν, εισάγουμε το 10ο και τελευταίο στοιχείο.

	0101
	0001
0000	0001
000	***01**

	0011
0010	0111
0010	0111
***10**	***11**

0100
100

6Η Άσκηση

Α) Η σημαντική πληροφορία είναι ότι έχω βάθος 10 στο ευρετήριο. Άρα έχω i = 10 bits και επομένως το ευρετήριο έχει 2^10 εγγραφές, δηλαδή 1024.

Β) Αφού έχω 1024 εγγραφές στο ευρετήριο και η κάθε εγγραφή κοστίζει 4 bytes, είναι αναμενόμενο το μέγεθος του ευρετηρίου να είναι 1024*4 δηλαδή 4096.

Γ) Ο μέγιστος αριθμός εγγραφών θα υπάρχει εάν έχουμε 1 κάδο για κάθε εγγραφή ευρετηρίου, δηλαδή 1024 κάδους. Ο κάθε κάδος χωράει 2400/400 = 6 εγγραφές και επομένως η μέγιστη περίπτωση είναι 1024*6 = 6144 εγγραφές.

7Η Άσκηση

Α) Δεν συμφωνώ με τη συνάρτηση κατακερματισμού που χρησιμοποίησε ο σχεδιαστής για τον εξής λόγο. Η υπόθεση ότι θα ακολουθεί την κανονική κατανομή η λύση του είναι λάθος, διότι είναι γεγονός ότι δεν θα υπάρχει καμία σχέση στον αριθμό των εγγραφών στο γκρουπ [1000,2000] με το γκρουπ [3000,4000] (στην Ελλάδα τουλάχιστον). Επομένως δεν είναι κατάλληλη η συνάρτηση κατακερματισμού που επέλεξε και άρα θα διαφωνήσω με τον σχεδιαστή.