ΠΛΗ 201 – Άσκηση 3

Στην παρούσα έκθεση περιγράφεται η γενική ιδέα επίλυσης της άσκησης 3, η παρουσίαση των μεθόδων που εφαρμόζει και η ερμηνεία των αποτελεσμάτων.

Ο κώδικας γράφτηκε στην βάση αυτού που μας δόθηκε στο φροντιστήριο του μαθήματος από το github.com και στις αλλαγές που προτάθηκαν από τους υπεύθυνους του εργαστηρίου γα την υλοποίηση του B+ -tree στο δίσκο.

searchRange()

Αποτελεί τη μέθοδο για την αναζήτηση εύρους τιμών. Αρχικά βρίσκει τη σελίδα δίσκου που πρέπει να είναι αποθηκευμένο το μικρότερο κλειδί της αναζήτησης. Αν το βρει το τυπώνει και συνεχίζει σε όλες τις σελίδες μέχρι να φτάσει στο μεγαλύτερο κλειδί της αναζήτησης. Αν δεν το βρει ψάχνει στις επόμενες σελίδες και τυπώνει όσα κλειδιά ανήκουν στο επιθυμητό εύρος τιμών.

search()

Μέθοδος που υλοποιεί την αναζήτηση κλειδιού στο δένδρο, ψάχνοντας στη σελίδα δίσκου που πρέπει να είναι αποθηκευμένο το κλειδί.

delete()

Μέθοδος που υλοποιεί τη διαγραφή διαγράφοντας το κλειδί από το δένδρο, ψάχνοντας στη σελίδα δίσκου που πρέπει να είναι αποθηκευμένο το κλειδί. Το διαγράφει και έπειτα διαχειρίζεται τους κόμβους που θέλουν επεξεργασία(Συγχώνευση ή δανεισμός από αδερφό).

insert()

Μέθοδος που υλοποιεί την εισαγωγή κόμβου στο δένδρο, τοποθετώντας το στην σωστή σελίδα δίσκου. Έπειτα διαχειρίζεται την πιθανή υπερχείλιση που μπορεί να συμβαίνει στον κόμβο που εισήχθη το κλειδί. Επίσης εισάγονται τα δεδομένα του κλειδιού σε ξεχωριστό αρχείο στο δίσκο.

Multicounter

Κλάση που δημιουργεί static int[10] και χρησιμοποιείται για την μέτρηση δεικτών απόδοσης, όπως παρουσιάσθηκε σε φροντιστήριο του μαθήματος.

ByteConverter

Κλάση που δοθέντος ένα int array το μετατρέπει σε byte array και το αντίστροφο.

Data

Κλάση που αποτελεί τα δεδομένα (data) που συνοδεύει κάθε κλειδί. Αποτελείται από 8 ακεραίους(ints), οι οποίοι μετατρέπονται σε δυαδική μορφή για την αποθήκευση τους στο δίσκο(32 bytes)

StorageCache

Βοηθητική κλάση για την διαχείριση του δένδρου. Μέσω αυτής γίνεται η ανάκτηση και εγγραφή των δεδομένων από και προς το δίσκο.

Test

Εκεί βρίσκεται η main μέθοδος μέσω της οποίας γίνεται η δημιουργία τυχαίων μοναδικών κλειδιών για εισαγωγή και αναζήτηση, η αρχικοποίηση των δομών δεδομένων και οι ζητούμενες αναζητήσεις .

<u>Αποτελέσματα</u>

Πίνακας 1: Αποτελέσματα μετρήσεων προσβάσεων δίσκου

Μέσος αριθμός προσβάσεων δίσκου ανά εισαγωγή	Μέσος αριθμός προσβάσεων δίσκου ανά τυχαία αναζήτηση*	Μέσος αριθμός προσβάσεων δίσκου ανά διαγραφή	Μέσος αριθμός προσβάσεων δίσκου για τυχαία αναζήτηση εύρους (K=10)*	Μέσος αριθμός προσβάσεων δίσκου για τυχαία αναζήτηση εύρους (K=1000)*
5,0	2,5	2,85	0,1	4,95

**Ο αριθμός προσβάσεων αφορά την εύρεση των κλειδιών στο αρχείο που αποθηκεύονται οι κόμβοι και δεν συμπεριλαμβάνει την ανάκτηση της αντίστοιχης σελίδας δίσκου από το αρχείο με τα πραγματικά δεδομένα.*

***Οι πληροφορίες αυτές εμφανίζονται και στην κονσόλα(ενδέχεται να διαφέρουν λόγω τυχαιότητας των παραγόμενων κλειδιών).*

<u>Ερμηνεία αποτελεσμάτων</u>

Από τα αποτελέσματα είναι προφανές πως ένα B+ -tree υλοποιημένο στο δίσκο απαιτεί ελάχιστες προσβάσεις σε αυτόν για την εκτέλεση των βασικών του λειτουργιών. Ιδιαίτερη προσοχή αξίζει να δοθεί στην αναζήτηση εύρους τιμών όπου ακόμα και για πολύ μεγάλο εύρος αναζήτησης οι προσβάσεις στο δίσκο είναι λίγες.

<u>Πηγές πληροφόρησης</u>

- https://github.com/linli2016/BPlusTree
- Προτεινόμενες αλλαγές από τα φροντιστήρια του μαθήματος
- https://stackoverflow.com/questions/11437203/how-to-convert-a-byte-array-to-an-int-array