

# ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΩΝ

## Αναφορά 3ης εργαστηριακής άσκησης

Χαϊντούτη Μαρία || AM 2020030129

Η main κλάση βρίσκεται στο πακέτο Main. Το πρόγραμμα μεταγλωττίζεται επιτυχώς, υλοποιώντας τις ζητούμενες λειτουργίες.

**Πακέτα:** Στον παραδοτέο κώδικα περιέχονται δύο πακέτα BTreeClasses και Main. Το πακέτο BTreeClasses περιέχει τις παρεχόμενες από την εκφώνηση κλάσεις BTree, BTreeLeafNode, BTreeInnerNode, BTreeNode, οι οποίες είναι υπεύθυνες για την δημιουργία του B+ Tree. Περιέχει, επίσης, τις κλάσεις Implementation, BPlusTreeMetrics, IndexInfo, οι οποίες είναι υπεύθυνες για την υλοποίηση των ζητούμενων λειτουργιών .

### **Περιγραφή κώδικα:**

- Στον έτοιμο κώδικα του B+ δέντρου οι μόνες παρεμβάσεις που έγιναν είναι η εισαγωγή μιας μεταβλητής public int M σε όλες τις κλάσεις στα κατάλληλα σημεία, η οποία αναπαριστά την τάξη του δένδρου. Επίσης, αυξάνονται οι προσβάσεις στον κόμβο και οι συγκρίσεις που γίνονται με κλειδιά του δένδρου. Οι προσβάσεις αυξάνονται στο τέλος του while της μεθόδου findLeafNodeShouldContainKey(TKey key) της BTree και στην αρχή της μεθόδου search στη BTreeLeafNode και οι συγκρίσεις στην μέθοδο search και της BTreeLeafNode κλάσης και της BTreeInnerNode μέσα στο for .
- **Implementation:**
  1. ProcessFile(String fileName) και readNextWord(RandomAccessFile raf): Ανοίγει ένα αρχείο κειμένου και επεξεργάζεται κάθε λέξη στο αρχείο. Εάν η λέξη υπάρχει ήδη στο B+ δέντρο, προσθέτει τις πληροφορίες θέσης και ονόματος αρχείου στην αντίστοιχη λίστα ευρετηρίου. Διαφορετικά, δημιουργεί μια νέα λίστα ευρετηρίου και εισάγει τη λέξη και τη λίστα στο B+ δέντρο. Η μέθοδος χρησιμοποιεί τη μέθοδο readNextWord, η οποία διαβάζει την επόμενη λέξη από ένα αρχείο RandomAccessFile.
  2. SearchWordInIndex(String searchWord): Αναζητά μια λέξη στο B+ δέντρο και εκτυπώνει τις πληροφορίες θέσης και ονόματος αρχείου, εάν η λέξη βρεθεί. Η μέθοδος χρησιμοποιεί την search του B+ δέντρου, για να εκτελέσει την αναζήτηση.
  3. PerformSearches(String fileName1, String fileName2): Δέχεται δύο ονόματα αρχείων ως όρισμα και εκτελεί τυχαίες αναζητήσεις στο B+ δέντρο χρησιμοποιώντας τις λέξεις από τα δύο αρχεία. Επιστρέφει έναν πίνακα με δύο στοιχεία, όπου το πρώτο στοιχείο είναι ο αριθμός πρόσβασης σε κόμβους και το δεύτερο στοιχείο είναι ο αριθμός συγκρίσεων που έγιναν κατά την αναζήτηση.
  4. GetRandomWordFromFile(String fileName, int size): Δέχεται ένα όνομα αρχείου και ένα μέγεθος ως όρισμα και επιστρέφει μια τυχαία επιλογή λέξεων από το αρχείο. Ανοίγει το αρχείο και διαβάζει κάθε γραμμή, χωρίζει τις λέξεις και τις προσθέτει σε μια λίστα. Στη συνέχεια, τυχαία επιλέγει λέξεις από τη λίστα με βάση το παραμετροποιημένο μέγεθος και τις επιστρέφει.

- BPlusTreeMetrics: Η κλάση χρησιμοποιείται για την καταγραφή και παρακολούθηση των επιδόσεων ενός B+ δέντρου, καθώς μπορεί να καταμετρά τον αριθμό των προσβάσεων σε κόμβους και τον αριθμό των συγκρίσεων που πραγματοποιούνται κατά τις αναζητήσεις και τις εισαγωγές στο δέντρο.
- IndexInfo: Η κλάση αντιπροσωπεύει πληροφορίες για τη θέση ενός στοιχείου στο ευρετήριο (index).
- Main: Η κλάση χρησιμοποιεί την κλάση Implementation για την υλοποίηση των λειτουργιών του B+ δέντρου και τις κλάσεις BPlusTreeMetrics και IndexInfo για την καταγραφή μετρήσεων και την αναπαράσταση των πληροφοριών του ευρετηρίου, αντίστοιχα. Αποτελείται από την μέθοδο menu() και ένα switch-case, για την εκτέλεση των ανάλογων λειτουργιών κάθε φορά.

A1. Μέσος αριθμός προσβάσεων κόμβων στο Ευρετήριο (B+-tree) για αναζήτηση Τάξη M=10	A2. Μέσος αριθμός συγκρίσεων με κλειδιά για αναζήτηση Τάξη M=10	B1. Μέσος αριθμός προσβάσεων κόμβων στο Ευρετήριο (B+-tree) για αναζήτηση Τάξη M=20	B2. Μέσος αριθμός συγκρίσεων με κλειδιά για αναζήτηση Τάξη M=20
4	18.16	3	25.95

- Από τους μέσους όρους των παραπάνω μετρήσεων παρατηρείται ότι, η τάξη M=20 παρουσιάζει καλύτερες επιδόσεις από την τάξη M=10 σε ό,τι αφορά τον αριθμό των προσβάσεων κόμβων και των συγκρίσεων. Αυτό συμβαίνει, επειδή μεγαλύτερες τιμές του M οδηγούν σε μεγαλύτερη χωρητικότητα του κάθε κόμβου και μειώνουν τον αριθμό των επιπέδων του δέντρου που πρέπει να αναζητηθούν για να βρεθεί η επιθυμητή λέξη. Αυτό επιφέρει μείωση του αριθμού των προσβάσεων κόμβων, βελτιώνοντας την απόδοση της αναζήτησης.
- Σε περίπτωση που τόσο το ευρετήριο όσο και το δέντρο αποθηκεύονταν στον δίσκο, θα έπρεπε το δέντρο να έχει δομή τέτοια που θα επιτρέπει την μείωση του αριθμού των προσβάσεων σε κόμβους του. Αυτό οφείλεται στο γεγονός ότι, η πρόσβαση στον δίσκο είναι πολύ αργή σε σχέση με την προσπέλαση της μνήμης, και έχει υψηλό κόστος. Συνεπώς, σε αυτήν την περίπτωση θα επιλεγόταν μια τάξη M=20, η οποία θα οδηγούσε σε λιγότερες προσβάσεις σε κόμβους και, κατά συνέπεια, λιγότερες προσβάσεις στη δομή του δέντρου και τον δίσκο. Ο αριθμός των προσβάσεων θα ήταν το κριτήριο απόφασης, καθώς η ανεπιθύμητη πρόσβαση στον δίσκο θα είχε σημαντικό αρνητικό αντίκτυπο στην απόδοση.