

# Δομές Δεδομένων και Αρχείων

Εργασία 2η: Υλοποίηση ενός *Btree* στην κεντρική μνήμη και στον δίσκο.

---

Σκοπός της εργασίας αυτής είναι να παρατηρήσουμε την αποδοτικότητα ενός ***Btree***, το οποίο έχει υλοποιηθεί στην κύρια μνήμη, καθώς επίσης και στην δευτερεύουσα (σκληρός δίσκος). Για την διευκόλυνση μας στην εκφώνηση προτείνεται η υλοποίηση του δέντρου στην κύρια μνήμη αρχικά και μετά στον δίσκο. Αυτή η ιδέα ακολουθήθηκε ουσιαστικά.

Το πρόγραμμα γράφτηκε σε γλώσσα προγραμματισμού ***Java***, στο περιβάλλον του ***eclipse***. Αποτελείται από τέσσερις κλάσεις, τις ***Node***, ***B\_tree***, ***DiskTree*** και ***Main***. Η κλάση ***Node*** αντιπροσωπεύει έναν οποιονδήποτε κόμβο του δέντρου με τα στοιχεία που περιλαμβάνει. Η κλάση ***B\_tree*** περιέχει την υλοποίηση του δέντρου και βασικών λειτουργιών του, στην κύρια μνήμη. Σε αυτή την κλάση περιέχονται και οι βασικές συναρτήσεις που ζητούνται και στην εκφώνηση:

- ***split***:

Η μέθοδος αυτή αναλαμβάνει να διασπάσει έναν κόμβο, εάν αυτός είναι γεμάτος και πρέπει να εισαχθεί σε αυτόν ένα νέο κλειδί. Η συνάρτηση είναι αναδρομική που σημαίνει ότι αν για παράδειγμα και ο “πατέρας” του κόμβου προς διάσπαση είναι γεμάτος, πρέπει και αυτός να διασπαστεί. Αν η διάσπαση πρέπει να γίνει στη ρίζα του δέντρου τότε η λειτουργία μετατίθεται σε μία βοηθητική μέθοδο, την ***splitRoot***, που αναλαμβάνει την διάσπαση της ρίζας.

- ***findNode***:

Αυτή η μέθοδος αναλαμβάνει να ανακαλύψει τον κόμβο του δέντρου, στον οποίο πρέπει να εισαχθεί ένα νέο δοσμένο κλειδί. Μετά την εύρεση του κόμβου αποφασίζεται από το πρόγραμμα αν πρέπει να γίνει διάσπαση ή εισαγωγή σε μη πλήρη κόμβο με την βοήθεια της συνάρτησης ***nonFullInsert***.

- ***insert***:

Η συνάρτηση αυτή αναλαμβάνει την εισαγωγή ενός νέου κλειδιού στο δέντρο. Αυτή η μέθοδος χρησιμοποιεί τις παραπάνω και ανάλογα με τον κόμβο, όπου πρέπει να εισαχθεί το κλειδί, καθώς και την κατάσταση του, αποφασίζει για τον τρόπο εισαγωγής του κλειδιού.

Αυτές είναι οι βασικές συναρτήσεις της κλάσης ***B\_tree***, αλλά υπάρχουν και άλλες βοηθητικές μικρές όπως η ***printTree***, για την εκτύπωση του δέντρου, ***fixLevel***, για την ενημέρωση του επιπέδου στο οποίο βρίσκεται κάποιος κόμβος κ.α.

## Υλοποίηση στο σκληρό δίσκο

---

Η κλάση ***B\_tree*** περιλαμβάνει την μέθοδο ***RandomizeTree*** η οποία παράγει τυχαία κλειδιά και τα εισάγει σε ένα δέντρο. Στην νέα κλάση λοιπόν, την ***DiskTree*** ανοίγουμε μια ροή προς ένα αρχείο και μεταφέρουμε το δέντρο από την κεντρική μνήμη στον δίσκο. Για τον σκοπό αυτό χρησιμοποιούνται κάποιες βασικές συναρτήσεις.

- ***writeTreeToDisk:***

αναλαμβάνει γενικά να πραγματοποιήσει αυτό που περιγράψαμε παραπάνω, δηλαδή την μεταφορά του δέντρου από την κύρια στην δευτερεύουσα μνήμη. Αυτό γίνεται με την βοήθεια και άλλων συναρτήσεων.

- ***loadBuffer:***

Η μέθοδος αυτή γεμίζει έναν buffer με τα στοιχεία μιάς σελίδας δίσκου του αρχείου.

- ***writeNodeToDisk:***

Σκοπός της μεθόδου είναι να μεταφέρει τα στοιχεία ενός κόμβου του δέντρου στο αρχείο.

- ***search:***

Ψάχνει τις σελίδες δίσκου μέχρι να βρει το κλειδί.

Για την υλοποίηση του δέντρου στον δίσκο βοήθησε η τεχνική της κατασκευής ενός δέντρου με βάση έναν πίνακα. Έτσι αν η θέση ενός κόμβου στο δίσκο είναι  $r$  τότε το πρώτο παιδί του θα είναι στην θέση  $order * r + 1$ , το δεύτερο παιδί στη θέση  $order * r + 2$  κτλ.

## Πηγές

---

- Σημειώσεις μαθήματος
- Το βιβλίο «Δομές Δεδομένων» του Παναγιώτη Δ. Μποζάνη

- **Web links:**

<http://en.wikipedia.org/wiki/B-tree>

<http://algs4.cs.princeton.edu/62btrees/BTree.java.html>

<http://slady.net/java/bt/view.php>

[http://read.pudn.com/downloads115/sourcecode/math/487532/Source%20Code/BTree.java\\_.htm](http://read.pudn.com/downloads115/sourcecode/math/487532/Source%20Code/BTree.java_.htm)

Τέλος Αναφοράς