**ΤΕΧΝΟΛΟΓΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**

**2019-2020**

Εργασία Υλοποίησης R\* - tree

Συμμετέχοντες :

Μάριος Νίκας

ΑΕΜ:3065

Παρμενίων Χαριστός

ΑΕΜ: 3173

Εισαγωγικά

Η εργασία έχει ως αντικείμενο την διαμόρφωση της δομής χωρικών δεδομένων R\*-tree η οποία ουσιαστικά είναι βελτιωμένη έκδοση της δομής R-tree. Η διαμόρφωση των τελικών αλγορίθμων που είναι υπεύθυνοι για την εκπόνηση των λειτουργιών της δομής έγινε με τη χρήση της προγραμματιστικής γλώσσας Java. Ο υπάρχον κώδικας εκτελεί τις βασικές λειτουργίες της δομής όπως είναι η insert και η split χρησιμοποιώντας χωρικά δεδομένα τα οποία δόθηκαν από τον Κύριο Παπαδόπουλο στη σελίδα του μαθήματος στο e-learning. Επιπλέον έγιναν δοκιμές και με άλλες βάσεις δεδομένων από διαφορετικές πηγές, όπως είναι το Open Street Maps, το μεγαλύτερο ηλεκτρονικό χάρτη ελεύθερης άδειας.

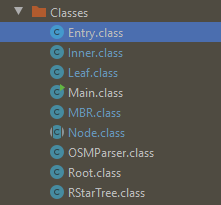
Η προσπάθεια για την εκτέλεση αυτής της εργασίας έγινε αξιοποιώντας αποκλειστικά το επιστημονικό άρθρο που παρείχε ο Κύριος Παπαδόπουλος. Βάση του pdf αυτού διαμορφώθηκε όλη η δομή του κώδικα και οι πολλαπλές κλάσεις. Εξάλλου το υλικό σχετικά με το R\*-tree στο διαδίκτυο είναι ελάχιστο.

Πρωτού αναφερθούμε στις λεπτομέρειες του κώδικα και της υλοποίησης πρέπει να ορίσουμε μερικές από τις παραμέτρους που πρέπει να ληφθούν υπόψη ως αναφορά την λειτουργία της δομής δεδομένων. Αρχικά ο κώδικας είναι διαμορφωμένος έτσι ώστε να υποστηρίζει δεδομένα πολλών διαστάσεων και όχι μόνο δυσδιάστατων στοιχείων. Επιπλέον το πρόγραμμα που έχουμε δημιουργήσει λαμβάνει τα δεδομένα μέσα από ένα ξεχωριστό αρχείο το οποίο λέγεται datafile. Από εκεί παίρνει δεδομένα σε blocks με μέγεθος 32kb.

Είναι κρίσιμο να αναφερθεί πως η υλοποίηση της δομής δεδομένων στην τρέχουσα κατάσταση της υποστηρίζει την βασική λειτουργία του δένδρου αλλά δεν υποστηρίζει ερωτήματα πολυδιάστατων δεδομένων όπως είναι τα ερωτήματα περιοχής και πλησιέστερων γειτόνων λόγω περιορισμένου χρόνου. Επιπλέον η εργασία έχει ως προϋπόθεση την διαμόρφωση ενός ξεχωριστού αρχείου το οποίο θα περιλαμβάνει τα στοιχεία του δένδρου και την οργάνωση των δεδομένων στου κόμβους του. Αυτή η λειτουργία δεν έχει υλοποιηθεί λόγω περιορισμένου χρόνου. Γι’ αυτό το λόγω δεν είναι δυνατή η καταγραφή στοιχείων για τη σύγκριση και δημιουργία διαφόρων γραφημάτων.

Ανάλυση Κώδικα

Ο κώδικας είναι γραμμένος σε java και είναι χωρισμένος σε πολλαπλές κλάσεις. Με αυτό το τρόπο γίνεται πιο δομημένος ο κώδικας και εκμεταλλευόμαστε την αντικειμενοστρεφή φύση της γλώσσας Java. Υπάρχουν 9 διαφορετικές κλάσεις οι οποίες συντελούν το πρόγραμμα του R\*-tree και αλληλοεπιδρούν μεταξύ τους. Αυτές είναι : Entry, Inner, Leaf, Main, MBR, Node, OSMParser, Root και RStarTree. Από το όνομα μπορεί κανείς να πάρει μια γενική ιδέα για το τι πραγματεύεται η κάθε κλάση πάραυτα στη συνέχεια θα αναλύσουμε τη κάθε κλάση ξεχωριστά.



**Entry Class**

Η κλάση Entry υλοποιεί τη μία εγγραφή ενός χωρικού δεδομένου. Έχει τις διαστάσεις καθώς και ένα id που δηλώνει την κάθε εγγραφή. Κάθε φορά που δημιουργείται νέο Entry παίρνει το πλήθος των διαστάσεων από τη Main και περνάει συντεταγμένες από το αρχείο που περιέχει τις εγγραφές μαζί με το id.

**Ιnner Class**

Η inner κλάση είναι υπεύθυνη για τους κόμβους του δένδρου που δεν είναι φύλλα. Κάνει extend την abstract κλάση Node και έχει ως δεδομένα (από κλάση Node) ένα Arraylist με τα Node παιδία, το id του ίδιου του κόμβου, ένα Node που δηλώνει το πατέρα, ένα αντικείμενο κλάσης MBR (minimum bounding rectangle) καθώς και έναν αριθμό που δηλώνει το επίπεδο/βάθος που βρίσκεται το συγκεκριμένο Node. Επιπλέον έχει δύο μεθόδους: η πρώτη παίρνει τα δεδομένα από τα παιδία του κόμβου και δημιουργεί minimum bounding rectangle και η δεύτερη επιστρέφει ένα αντικείμενο Node που είναι αντίγραφο του κόμβου που κάλεσε αυτή τη μέθοδο.

**Leaf Class**

Η συγκεκριμένη κλάση φτιάχνει την λειτουργεία του φύλλου. Είναι παρόμοια με την προηγούμενη κλάση καθώς κάνει και αυτή extend τη κλάση Node. Αντί για παιδία Node έχουμε ένα Arraylist με Entries και επιπρόσθετα το επίπεδο αρχικοποιείται ως 1. Όπως και στη κλάση Inner υπάρχουν δύο μέθοδοι που κάνουν το ίδιο πράγμα, η μια δημιουργεί το MBR και η άλλη επιστρέφει ένα αντίγραφο του φύλλου.

**Node Class**

Η κλάση Node είναι μία abstract κλάση η οποία περιέχει τις μεταβλητές που χρησιμοποιούνται στις δύο προηγούμενες κλάσεις και τις δύο κοινές μεθόδους. Επιπλέον υπάρχει η function assignID που αυξάνει και επιστρέφει το blockID το οποίο είναι static ακέραιος.

**Root class**

Ουσιαστικά αποτελεί κλάση που αναπαριστά την ρίζα του δένδρου. Όπως και η Leaf και η Inner, αυτή η κλάση κάνει extend τη Node και έχει τις ίδιες μεταβλητές. Δεν υποστηρίζει την μέθοδο formMBR.

**OSMParser Class**

Αυτή η κλάση διαχειρίζεται ότι έχει να κάνει με την διαμόρφωση και την ανάγνωση των αρχείων που συνοδεύουν τον κώδικα του project. Από τον κατασκευαστή η κλάση παίρνει το αρχείο osm και αρχικοποιεί ένα αντικείμενο File που περιέχει τις τελικές εγγραφές. Η μέθοδος parse ουσιαστικά παίρνει τα δεδομένα από το αρχείο osm και τα αποθηκεύει στο datafile. Πρώτα όμως τα διαμορφώνει με τέτοιο τρόπο που να μπορέι ο υπόλοιπος κώδικας μας να τα επεξεργαστεί. Σε αυτή τη διαδικασία βοηθάει η μέθοδος clean η οποία Η clean στο OSMParser παίρνει ως όρισμα ένα string που περιέχει ενα attribute του entry σε μορφή id=“63829173” και επιστρέφει τα δεδομένα που είναι μέσα στα “ “

**MBR Class**

H MBR κλάση αποτελεί την υλοποίηση του minimum bounding rectangle το οποίο είναι ένα n-διάστατο ορθογώνιο που περιέχει τα παιδία του κάθε κόμβου ή τις εγγραφές αν ο κόμβος είναι φύλο. Ο δυσδιάστατος πίνακας Edges αρχικοποιείται στο κατασκευαστή και έχει n = πλήθος διαστάσεων γραμμές και 2 στήλες. Σε κάθε γραμμή του πίνακα αυτού του πίνακα αποθηκεύουμε την ελάχιστη και μέγιστη τιμή που έχει το ορθογώνιο για κάθε διάσταση. Η μεταβλητή area αποθηκεύει το εμβαδόν του ορθογωνίου και ο n-διάστατος πίνακας center περιέχει τις συντεταγμένες του κέντρου του ορθογωνίου. Με την μέθοδο calculateArea αναβαθμίζεται το εμβαδό, ενώ με την getCenter επιστρέφεται το κέντρο του MBR. Η μέθοδος calculateOverlap υπολογίζει τη συνολική επικάλυψη του υπάρχοντος MBR με ένα άλλο που δίνεται ως όρισμα χρησιμοποιώντας τον αλγόριθμο….. Τέλος η calculateMargin υπολογίζει και επιστρέφει τη περίμετρο του σχήματος. Όλες αυτές οι μέθοδοι χρησιμοποιούνται στις πολύπλοκες λειτουργίες του R\*-tree (πχ Reinsert).

**RStarTree Class**

Η κλάση RStarTree είναι εκεί που πραγματοποιείται όλη η διεργασία του δένδρου. Έχει μία ρίζα (root), μία λίστα από όλα τα Node του δένδρου, δύο μεταβλητές Μ και m που δηλώνουν το μέγιστο και το ελάχιστο πλήθος εγγραφών/παιδιών που μπορεί να χωράει ένας κόμβος καθώς και το συνολικό βάθος του δένδρου. Οι περισσότεροι μέθοδοι που υλοποιούνται στο κώδικα είναι η αντίστοιχη κωδικοποίηση των αλγορίθμων που περιγράφονται στο επιστημονικό άρθρο. Η chooseSubTree διαλέγει το κατάλληλο Node που θα εισαχθεί ένα Leaf. Η overflowTreatment καλέι την reinsert άμα υπερχείλιση στο κόμβο που δίνεται ως όρισμα, αλλιώς καλεί την split. H reinsert βοηθάει στην βελτιστοποίηση της απόδοσης του δένδρου με το να αφαιρεί και να ξανά εισάγει τις εγγραφές σε ένα κόμβο. Η euclidean υπολογίζει την απόσταση μεταξύ δύο σημείων (κέντρων MBR) και χρησιμοποιείται στη reinsert.

**Main Class**

Τέλος η Main αρχικοποιεί τις διαστάσεις και καλεί την κλάση OSMParser με παράμετρο το αρχείο OSM για να ξεκινήσει η άντληση των δεδομένων.

Συμπέρασμα

Η εργασία, όπως είναι διαμορφωμένη από εμάς υλοποιεί την δομή του R\*-tree και δεν υποστηρίζει χωρικά ερωτήματα όπως είναι το ερώτημα των κοντινότερων πρακτόρων. Τελικά ο χρόνος αποδείχθηκε περιορισμένος για εμάς και γι’ αυτό η εργασία δεν βρίσκεται στο 100%. Πάραυτα ο υπάρχον κώδικας είναι καλό “proof of concept” για την επέκταση και ολοκλήρωση του project. Δυστυχώς δεν μπορούμε να παρέχουμε κανένα διάγραμμα γιατί δεν έχει τρέξει κανένα ερώτημα πάνω στο δένδρο μας.