Εργασία 2023 - 2024

Τεχνολογια βασεων δεδομενων

Υλοποίηση Χωρικών Μεθόδων Προσπέλασης

Ναλμπάντη παναγιωτα, 4050

παντελοπουλου βασιλικη, 4153

Περιεχόμενα

[1. ΕΙΣΑΓΩΓΉ 2](#_Toc174532965)

[2. ΜΕΘΟΔΟΛΟΓΙΑ 3](#_Toc174532966)

[3. ΥΛΟΠΟΙΗΣΗ R\* - TREE 3](#_Toc174532967)

[3.1. Δημιουργία και διαχείριση datafile 3](#_Toc174532968)

[3.2. Δημιουργία του R\* - tree 3](#_Toc174532969)

[3.3. Δημιουργία και διαχείριση indexfile 4](#_Toc174532970)

[3.4. Εισαγωγή εγγραφής 4](#_Toc174532971)

[3.4.1. Υλοποίηση 4](#_Toc174532972)

[3.4.2. Εκτέλεση παραδείγματος 4](#_Toc174532973)

[3.5. Διαγραφή εγγραφής 4](#_Toc174532974)

[3.5.1. Υλοποίηση 4](#_Toc174532975)

[3.5.2. Εκτέλεση παραδείγματος 4](#_Toc174532976)

[4. ΕΡΩΤΗΜΑΤΑ ΠΑΝΩ ΣΤΟ R\* - TREE 4](#_Toc174532977)

[4.1. Ερώτημα περιοχής (Range query) 4](#_Toc174532978)

[4.1.1. Υλοποίηση 4](#_Toc174532979)

[4.1.2. Εκτέλεση παραδείγματος 4](#_Toc174532980)

[4.2. Ερώτημα k πλησιέστερων γειτόνων (KNN) 4](#_Toc174532981)

[4.2.1. Υλοποίηση 4](#_Toc174532982)

[4.2.2. Εκτέλεση παραδείγματος 4](#_Toc174532983)

[4.3. Ερώτημα κορυφογραμμής (Skyline) 4](#_Toc174532984)

[4.3.1. Υλοποίηση 4](#_Toc174532985)

[4.3.2. Εκτέλεση παραδείγματος 4](#_Toc174532986)

[5. ΜΑΖΙΚΗ ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΔΕΝΔΡΟΥ (BOTTOM – UP) 4](#_Toc174532987)

[5.1. Υλοποίηση 4](#_Toc174532988)

[6. ΠΑΡΑΤΗΡΗΣΕΙΣ 4](#_Toc174532989)

## 1. ΕΙΣΑΓΩΓΉ

Η εργασία αυτή ασχολείται με την οργάνωση και την επεξεργασία δεδομένων χώρου με πολλαπλές διαστάσεις. Συγκεκριμένα, γίνεται υλοποίηση της κατασκευής μίας δομής δεδομένων δευτερεύουσας μνήμης, του R\*- tree. Αυτή υποστηρίζει βασικά ερωτήματα (range query, knn, skyline), καθώς και κατασκευή καταλόγου με τις βασικές λειτουργίες του.

Για την υλοποίηση των παραπάνω χρησιμοποιήσαμε το δοθέν αρχείο osm, το οποίο περιέχει δεδομένα από το OpenStreetMap που αναφέρονται σε περιοχές του κόσμου με σημεία ενδιαφέροντος. Είναι απαραίτητο να αντιμετωπίσουμε τα δεδομένα ως δισδιάστατα, καθώς τα δεδομένα από το osm αρχείο έχουν μόνο συντεταγμένες lat και lon. Όλες οι υπόλοιπες συναρτήσεις του project λειτουργούν θεωρώντας ότι μπορούμε να έχουμε περισσότερες από δύο διαστάσεις. Επιλέχθηκε η εκπόνησή της με γλώσσα Python, διότι παρέχει πολλές χρήσιμες βιβλιοθήκες, οι οποίες δεν χρειάζονται εγκατάσταση καθώς ανήκουν στην Python Standard Library. Σε όλο το project χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

* xml.etree.ElementTree
* heapq
* math
* time
* sys

Για την αποθήκευση και αναπαράσταση των εγγραφών χρησιμοποιήθηκε μορφή xml, καθώς παρέχει έναν εύκολα διαχειρίσιμο και πλήρως κατανοητό τρόπο αποθήκευσης δεδομένων, ενώ ταυτόχρονα επιτρέπει το read και write ανά block, το οποίο ήταν το κύριο ζητούμενο από το datafile ώστε να προσομοιωθεί η λειτουργία τους στην μνήμη/buffer. Το μόνο μειονέκτημα της xml αναπαράστασης είναι η απόδοσή του καθώς, όπως είναι γνωστό, δεν φημίζεται για την ταχύτητά του. Παρόλα αυτά, εφόσον εξετάζονται χρονικά οι διεργασίες της δομής R\*-Tree ως δευτερεύον κατάλογος, και η ταχύτητα εκτέλεσης των διάφορων queries δεν επηρεάζονται από την αναπαράσταση αυτή, κρίθηκε ως αποδεκτό format για το datafile, αφού επηρεάζει μόνο την ταχύτητα κατασκευής του καταλόγου. Ένα ενδεικτικό τμήμα ενός datafile.xml αρχείου δίνεται παρακάτω :

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

## 2. ΜΕΘΟΔΟΛΟΓΙΑ

Πριν την έναρξη της κατασκευής του R\* δένδρου, χρειάζεται η δημιουργία μιας λίστας, η οποία θα αποτελείται από ένα πλήθος blocks των 32KB. Αυτά θα περιέχουν λίστες από records, τα οποία λαμβάνουμε από το osm αρχείο. Τα records περιέχουν πληροφορίες σχετικά με το αναγνωριστικό, το όνομα και τις συντεταγμένες ενός σημείου (id, name, lat/ lon/ κλπ.). Οι διάφορες συναρτήσεις βρίσκουν τον αριθμό των διαστάσεων μέσα από τα δεδομένα. Επομένως, αν θέλουμε η δομή να λειτουργήσει για δεδομένα πολλών διαστάσεων αρκεί να δώσουμε στην συνάρτηση *createBlocks()* μια λίστα *record\_data[]*, η οποία θα περιέχει records της μορφής: *[ id, name, coord\_1, coord\_2, …, coord\_n ]*.

Αυτό θα έχει ως αποτέλεσμα να δημιουργηθεί το επιθυμητό datafile που περιέχει σημεία πολλαπλών διαστάσεων.

Συνεχίζουμε δημιουργώντας ένα αρχείο datafile.xml, το οποίο θα περιέχει την λίστα των blocks με τα records σε μορφή xml.

## 3. ΥΛΟΠΟΙΗΣΗ R\* - TREE

### 3.1. Δημιουργία και διαχείριση datafile

Ο κώδικας που υλοποιήθηκε στο ***create\_datafile.py*** παράγει ως αποτέλεσμα το αρχείο ***datafile.xml***. Για την ευκολότερη διαχείριση των δεδομένων και την δημιουργία του datafile χρησιμοποιήθηκε η βιβλιοθήκη της Python *xml.etree.ElementTree* για την ανάγνωση του αρχικού osm αρχείου. Πιο αναλυτικά, μετά την λήψη του osm αρχείου και αρχικοποιώντας μία λίστα *record\_data[]*, προστέθηκαν προσωρινά σε αυτή μόνο τα δεδομένα του osm με ετικέτα ‘node’ εφόσον μας ενδιαφέρουν μόνο τα σημεία. Από κάθε node διατηρούνται μόνο το **id**, το **όνομα (name)** και οι **lat-lon** συντεταγμένες του. Σε περίπτωση που κάποιο node δεν διαθέτει όνομα (tag name), ονομάζεται ‘unknown’. Αν τα σημεία προσδιορίζονται από περισσότερες διαστάσεις και όχι μόνο lat-lon, αυτές μπορούν να προστεθούν σε αυτό το σημείο: *record\_data.append([id, name, lat, lon])*.

Έπειτα, για την αποθήκευση των εγγραφών σε blocks των **32KB**, η λίστα με τα δεδομένα (*record\_data[]*) δίνεται ως παράμετρος στην συνάρτηση ***createBlocks(record\_data)***, η οποία είναι υπεύθυνη για την δημιουργία των blocks μεγέθους 32ΚΒ, που προσδιορίζεται από το block\_size = 32 \* 1024. Για να αναπαρασταθεί ο διαχωρισμός των εγγραφών σε blocks επιστρέφεται μία λίστα από λίστες *listOfBlocks[]*. Στη πρώτη θέση της λίστας βρίσκεται το **block0**, το οποίο περιέχει βοηθητικές πληροφορίες σχετικά με το πλήθος των εγγραφών και τον αριθμό των blocks που δημιουργήθηκαν.

Μετά την διαδικασία διαχωρισμού των εγγραφών σε blocks, αυτά αποθηκεύονται στο αρχείο datafile.xml μέσω της συνάρτησης ***create\_xml(blocks, records, xml)***, που δέχεται ως παραμέτρους τη λίστα με τα blocks, το πλήθος των εγγραφών και το όνομα του αρχείου (προαιρετικά).

### 3.2. Δημιουργία του R\* - tree

### 3.3. Δημιουργία και διαχείριση indexfile

### 3.4. Εισαγωγή εγγραφής

#### 3.4.1. Υλοποίηση

#### 3.4.2. Εκτέλεση παραδείγματος

### 3.5. Διαγραφή εγγραφής

#### 3.5.1. Υλοποίηση

#### 3.5.2. Εκτέλεση παραδείγματος

## 4. ΕΡΩΤΗΜΑΤΑ ΠΑΝΩ ΣΤΟ R\* - TREE

### 4.1. Ερώτημα περιοχής (Range query)

#### 4.1.1. Υλοποίηση

#### 4.1.2. Εκτέλεση παραδείγματος

### 4.2. Ερώτημα k πλησιέστερων γειτόνων (KNN)

#### 4.2.1. Υλοποίηση

#### 4.2.2. Εκτέλεση παραδείγματος

### 4.3. Ερώτημα κορυφογραμμής (Skyline)

#### 4.3.1. Υλοποίηση

#### 4.3.2. Εκτέλεση παραδείγματος

## 5. ΜΑΖΙΚΗ ΚΑΤΑΣΚΕΥΗ ΤΟΥ ΔΕΝΔΡΟΥ (BOTTOM – UP)

### 5.1. Υλοποίηση

## 6. ΠΑΡΑΤΗΡΗΣΕΙΣ