ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017-18)

ΕΡΓΑΣΙΑ 2 – Χωρικά Δεδομένα

Προθεσμία: 20 Απριλίου 2018, 9μ.μ.

Στόχος της εργασίας είναι η ανάπτυξη τεχνικών δεικτοδότησης (δηλ. ευρετηρίασης) και αναζήτησης χωρικών δεδομένων.

Μέρος 1 (προγραμματιστικό)

Κατεβάστε το αρχείο Beijing_restaurants.txt από το ecourse. Το αρχείο περιέχει τις συντεταγμένες 51970 σημείων τα οποία είναι θέσεις εστιατορίων στο Πεκίνο. Η πρώτη γραμμή του αρχείου είναι το πλήθος των σημείων και κάθε άλλη γραμμή οι x και y συντεταγμένες ενός εστιατορίου.

Γράψτε ένα πρόγραμμα το οποίο θα υλοποιεί την τεχνική sort-tile-recursive (STR) για να διαβάσει τα δεδομένα από το αρχείο και να φτιάξει ένα R-tree στη μνήμη για αυτά. Το δέντρο σας θα πρέπει να έχει δύο τύπων κόμβους: φύλλα και μη-φύλλα. Τα φύλλα αποθηκεύουν εγγραφές τύπου <record-id, σημείο> ενώ οι ενδιάμεσοι κόμβοι εγγραφές του τύπου <node-id, MBR>. Θεωρήστε ότι το record-id αντιστοιχεί στη γραμμή στην οποία βρίσκεται το αντίστοιχο σημείο στο αρχείο. Δηλαδή το εστιατόριο με συντεταγμένες 39.856138,116.42394 έχει record-id 1, το εστιατόριο με συντεταγμένες 39.813336,116.486149 έχει record-id 2, κλπ. Θεωρήστε ότι (α) ο κάθε κόμβος έχει χωρητικότητα 1024 bytes και ότι χρησιμοποιούμε 8 bytes για κάθε συντεταγμένη και 4 bytes για κάθε node-id ή record-id. Άρα κάθε σημείο χρειάζεται 16 bytes για τις συντεταγμένες και κάθε MBR 32 bytes.

Στην υλοποίησή σας θα πρέπει να διαβάσετε τα δεδομένα από το αρχείο, να κάνετε την ταξινόμηση και να φτιάξετε το δέντρο στη μνήμη. Θα χρησιμοποιήσετε μια δομή array ή vector για να αποθηκεύσετε τους κόμβους. Καθώς φτιάχνετε το δέντρο, θα προσθέτετε τους κόμβους στο array ή vector και το node-id ενός κόμβου θα είναι η θέση του στο array ή vector. Με αυτό τον τρόπο προσομοιώνουμε μια ακολουθία από blocks στο δίσκο που αποθηκεύουν το δέντρο.

Το πρόγραμμά σας **θα πρέπει να τυπώνει στατιστικά για το δέντρο**: ύψος, αριθμός κόμβων σε κάθε επίπεδο, και μέσο εμβαδό των MBRs σε κάθε επίπεδο (στο επίπεδο των φύλλων το μέσο εμβαδό είναι 0, γιατί τα φύλλα αποθηκεύουν σημεία). Επίσης θα πρέπει να γράφει σε ένα text αρχείο εξόδου tree.csv (CSV = comma separated values) την αναπαράσταση του

δέντρου. Η πρώτη γραμμή θα πρέπει να έχει μόνο το node-id της ρίζας του δέντρου. Κάθε γραμμή του αρχείου θα περιέχει τα δεδομένα ενός κόμβου στην εξής μορφή:

node-id, n, f, (ptr1, geo1), (ptr2, geo2), …, (ptrn, geon)

όπου node-id είναι το id του κόμβου, n ο αριθμός των εγγραφών στον κόμβο, f είναι 0 ή 1 ανάλογα με το αν ο κόμβος είναι φύλλο ή όχι, και ακολουθούν οι n εγγραφές μέσα στον κόμβο. Σε κάθε εγγραφή, το ptr είναι είτε ένα node-id (αν η εγγραφή δείχνει σε ενδιάμεσο κόμβο) είτε ένα record-id αν η εγγραφή δείχνει σε αντικείμενο. Το geo είναι είτε μια ακολουθία 2 αριθμών αν πρόκειται για σημείο, ή 4 αριθμών αν πρόκειται για MBR.

Το πρόγραμμά σας θα πρέπει να τρέχει στη γραμμή διαταγών και να παίρνει σαν ορίσματα το αρχείο με τα δεδομένα και το όνομα του αρχείου εξόδου.

Για τις λεπτομέρειες του STR bulk loading μπορείτε να ανατρέξετε στις σημειώσεις του μαθήματος ή στο παρακάτω άρθρο:

https://pdfs.semanticscholar.org/e680/839472e7f5cab0f12fcc7c4aba6834a2e096.pdf

Χρησιμοποιήστε έτοιμες συναρτήσεις ταξινόμησης στη μνήμη (όχι external sorting).

Μέρος 2 (προγραμματιστικό)

Υλοποιήστε τον incremental nearest neighbor search αλγόριθμο που βασίζεται σε best-first search. Εφαρμόστε τον στο δέντρο που έχετε φτιάξει και αποθηκεύσει στο αρχείο εξόδου από το 1ο μέρος της άσκησης.

Γράψτε ένα πρόγραμμα το οποίο θα παίρνει σαν όρισμα από τη γραμμή διαταγών το αρχείο του δέντρου, τις συντεταγμένες ενός σημείου αναφοράς q και έναν αριθμό k και υπολογίζει και θα τυπώνει τους k πλησιέστερους γείτονες του q, τους k+1 πλησιέστερους γείτονες και τους k+2 πλησιέστερους γείτονες αυξητικά. Το πρόγραμμα θα πρέπει να τυπώνει το σημείο (id και συντεταγμένες) που είναι ο αντίστοιχος γείτονας καθώς και τα περιεχόμενα της ουράς προτεραιότητας μόλις έχει υπολογιστεί ο γείτονας. Για παράδειγμα τα 5 πλησιέστερα σημεία στο σημείο με συντεταγμένες (39.7,116.5) και οι αποστάσεις τους είναι τα:

(18883, 0.004392425412003902)
(50630, 0.00955500000000051)
(8962, 0.009651267274305202)
(12977, 0.014737041833420804)
(11336, 0.016285383016681726)

Το αρχείο στο οποίο έχει αποθηκευτεί το δέντρο θα πρέπει να διαβαστεί εξ ολοκλήρου και να φτιάξετε μια αναπαράσταση του δέντρου στη μνήμη όπως και στο 1ο μέρος της άσκησης πριν αρχίσετε να ψάχνετε στο δέντρο. Θα χρειαστεί να φτιάξετε μια συνάρτηση mindist η οποία υπολογίζει την (ελάχιστη) Ευκλείδεια απόσταση μεταξύ ενός σημείου (του q) και ενός MBR. Η απόσταση μπορεί να συντεθεί από τις αποστάσεις σε κάθε άξονα όπως έχουμε πει στο μάθημα. Δηλαδή είναι η ρίζα του αθροίσματος των τετραγωνισμένων αποστάσεων σε κάθε διάσταση.