ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων (ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2020-21)

ΕΡΓΑΣΙΑ 2 – Χωρικά Δεδομένα (προθεσμία: 28 Απριλίου 2021, 9μ.μ.)

Στόχος της εργασίας είναι η κατασκευή και η χρήση ενός ευρετηρίου R-tree για χωρικά δεδομένα.

Μέρος 1: Κατασκευή R-Tree μέσω Bulk Loading

Σας ζητείται να γράψετε ένα πρόγραμμα το οποίο θα κατασκευάζει ένα R-Tree για ένα σύνολο από πολύγωνα, τα οποία θα τα διαβάσετε από δυο αρχεία κειμένου.

Το πρόγραμμά σας θα πρέπει να υπολογίζει τα MBRs των αντικειμένων και μετά να κάνει bulk loading το δέντρο αφού ταξινομήσει τα MBRs με τη χρήση μιας καμπύλης πλήρωσης χώρου (space-filling curve), πιο συγκεκριμένα την z-order. Αυτές οι καμπύλες είναι συναρτήσεις, οι οποίες αντιστοιχίζουν (map) ένα πολυδιάστατο διάνυσμα σε έναν αριθμό στον μονοδιάστατο χώρο. Δύο αντικείμενα που βρίσκονται κοντά χωρικά έχουν μεγάλη πιθανότητα να αντιστοιχιστούν σε κοντινές τιμές. Κατ' επέκταση δύο MBRs που είναι κοντά στο χώρο θα μπουν με μεγάλη πιθανότητα στο ίδιο φύλλο του δέντρου.

Περισσότερες πληροφορίες μπορείτε να βρείτε στις σημειώσεις του μαθήματος ή στους παρακάτω συνδέσμους:
https://en.wikipedia.org/wiki/Space-filling_curve
https://en.wikipedia.org/wiki/Z-order_curve

Σας δίνονται δυο αρχεία εισόδου, τα **coords.txt** και **offsets.txt**. Το πρώτο περιέχει συντεταγμένες σημείων στη μορφή <x>,<y>. Το δεύτερο περιέχει εγγραφές της μορφής <id>,<startOffset>,<endOffset> όπου id είναι το μοναδικό αναγνωριστικό ενός πολυγωνικού αντικειμένου και startOffset (αντίστοιχα endOffset) είναι ο αρ. γραμμής στο αρχείο coords.txt όπου αρχίζουν (αντίστοιχα τελειώνουν) οι συντεταγμένες των σημείων που σχηματίζουν το κάθε αντικείμενο.

Διαβάζοντας τα δυο αρχεία, βρείτε τις συντεταγμένες για κάθε αντικείμενο και στη συνέχεια το minimum bounding rectangle (MBR) του κάθε αντικειμένου. Μετασχηματίστε το κέντρο κάθε MBR για να υπολογίσετε την z-order curve τιμή που αντιστοιχεί στο MBR. Μετά κάντε ταξινόμηση με βάση τις z-order τιμές και «πακετάρετε» τα MBRs σε φύλλα και αναδρομικά φτιάξτε τα επόμενα επίπεδα του R-tree μέχρι τη ρίζα (*για τα επίπεδα πάνω από τα φύλλα δεν χρειάζεται να ξανακάνετε z-order ταξινόμηση*). Για την κατασκευή του δέντρου θεωρήστε ότι κάθε κόμβος του δέντρου έχει χωρητικότητα **20** και ελάχιστο αριθμό από εγγραφές 0.4*20 = **8**. Αυτό σημαίνει ότι αν ο τελευταίος κόμβος του κάθε επιπέδου έχει λιγότερες από 8 εγγραφές, τότε πρέπει να προσαρμόσετε τον αριθμό των εγγραφών ώστε να έχει ακριβώς 8 εγγραφές και ο προηγούμενος να έχει λιγότερες από 20. Εξαίρεση αποτελεί η ρίζα του δέντρου η οποία μπορεί να έχει από 2 μέχρι 20 παιδιά. Τα φύλλα του δέντρου περιέχουν εγγραφές του τύπου [id, MBR] όπου MBR είναι το MBR του αντικειμένου με αναγνωριστικό id στη μορφή [x-low, x-high, y-low, y-high]. Οι κόμβοι που δεν είναι φύλλα περιέχουν εγγραφές του τύπου [id, MBR] όπου id είναι το αναγνωριστικό του κόμβου (node-id) που δείχνει η εγγραφή και MBR είναι το MBR όλων των εγγραφών αυτού του κόμβου στη μορφή [x-low, x-high, y-low, y-high].

Για να μετασχηματίσετε ένα ζευγάρι (x,y) συντεταγμένων σε έναν κωδικό z-order μπορείτε να χρησιμοποιήσετε τη συνάρτηση interleave_latlng από την παρακάτω βιβλιοθήκη (ή να την γράψετε στη γλώσσα της επιλογής σας):
https://github.com/trevorprater/pymorton/blob/master/pymorton/pymorton.py
Σε αυτή τη συνάρτηση lat=latitude=γεωγραφικό πλάτος=y συντενταγμένη και lng=longitude=γεωγραφικό μήκος=x συντενταγμένη, οπότε την καλείτε με interleave_latlng(y,x) για να σας βγάλει μία z-τιμή σε μορφή αλφαριθμητικού. Ταξινομήστε με βάση αυτά τα αλφαριθμητικά.

Το πρόγραμμά σας θα πρέπει να παίρνει σαν ορίσματα (command-line arguments) τα δυο αρχεία που σας δίνονται.

Στην έξοδο του προγράμματος, τυπώστε τον αριθμό των κόμβων ανά επίπεδο στο δέντρο, για παράδειγμα:

```
500 nodes at level 0
25 nodes at level 1
2 nodes at level 2
1 node at level 3
```

Το πρόγραμμα θα πρέπει να γράφει στο αρχείο εξόδου **Rtree.txt** όλους τους κόμβους του δέντρου με τη σειρά των ids τους. Η μορφή κάθε γραμμής πρέπει να είναι:

[isnonleaf, node-id, [[id1, MBR1], [id2, MBR2], …, [idn, MBRn]]]

όπου isnonleaf=1 αν ο κόμβος δεν είναι φυλλο (isnonleaf=0 αν είναι φύλλο), node-id είναι το id του κόμβου και ακολουθούν οι εγγραφές μέσα στον κόμβο. Σε κάθε εγγραφή, το id είναι είτε ένα node-id (αν η εγγραφή δείχνει σε κόμβο) είτε ένα object-id αν η εγγραφή δείχνει σε αντικείμενο. Το MBR, τύπου [x-low, x-high, y-low, y-high], είναι είτε το MBR ενός κόμβου (αν isnonleaf=1) ή ενός αντικειμένου (αν isnonleaf=0). Για παράδειγμα η πρώτη γραμμή του αρχείου είναι:

```
[0, 0, [[5868, [-170.844179, -170.707084, -14.373776, -14.287277]],
..., [3060, [-157.850181, -157.848054, 21.301518, 21.303834]]]]
```

Ο κόμβος στην τελευταία γραμμή του αρχείου θα πρέπει να είναι η ρίζα.

Μέρος 2: Ερωτήσεις Εύρους (Range queries)

Υλοποιήστε μια συνάρτηση αποτίμησης ερωτήσεων εύρους στο R-Tree που φτιάξατε. Το εύρος της ερώτησης καθορίζεται από ένα ορθογώνιο W και το ζητούμενο είναι να βρεθούν τα MBRs που τέμνουν το W. Η συνάρτηση σας λοιπόν θα παίρνει σαν όρισμα το nodeid της ρίζας του R-tree, το ορθογώνιο-ερώτηση W στη μορφή [x-low, x-high, y-low, y-high] και θα υπολογίζει τα αποτελέσματα της ερώτησης χρησιμοποιώντας το R-tree. Για να ελέγξετε τη συνάρτηση σας δίνεται το αρχείο Rqueries.txt που περιέχει ερωτήσεις της μορφής

<x_low> <y_low> <x_high> <y_high>

Όπου x-low είναι το κάτω όριο της x διάστασης του παραθύρου ερώτησης W και x-high το άνω όριο. Αντίστοιχα για την y διάσταση. Για κάθε ερώτηση τυπώστε τα ids των αντικειμένων των οποίων το MBR επικαλύπτεται με το παράθυρο W της ερώτησης (filter step).

Το πρόγραμμά σας θα πρέπει να παίρνει σαν command-line arguments το αρχείο του R-tree που δημιουργήσατε στο Μέρος 1 και το αρχείο των ερωτήσεων Rqueries.txt. Θα πρέπει να δημιουργεί το δέντρο από τα περιεχόμενα του αρχείου του R-tree και κατόπιν να διαβάζει

και να εκτελεί διαδοχικά τις ερωτήσεις από αρχείο ερωτήσεων. Για κάθε ερώτηση πρέπει να τυπώνει στην έξοδο τη γραμμή της ερώτησης (αρχίζοντας από τη γραμμή 0), τον αριθμό των αποτελεσμάτων σε παρένθεση, και τα ids των αποτελεσμάτων, δηλαδή των αντικειμένων που περνάνε από το filter step.

Παράδειγμα:

```
0 (7): 2527,2712,8371,5042,7080,7656,7944
...
```

Μέρος 3: Ερωτήσεις Πλησιέστερου Γείτονα (kNN queries)

Υλοποιήστε τον αλγόριθμο best-first search για ανάκτηση του πλησιέστερου object MBR σε ένα σημείο αναφοράς q. Η έκδοση του αλγορίθμου θα πρέπει να είναι εκείνη που κάνει incremental NN search και χρησιμοποιεί μία ουρά προτεραιότητας για να οργανώνει τα πλησιέστερα R-tree node entries που έχει δει σε μία ουρά προτεραιότητας. Η ουρά προτεραιότητας θα πρέπει να περιλαμβάνει node MBRs αλλά και object MBRs. Όταν ένα object MBR βγαίνει από την ουρά είναι και το επόμενο πλησιέστερο MBR στο q.

Καλείστε να γράψετε μία συνάρτηση, η οποία παίρνει σαν όρισμα το id της ρίζας του δέντρου και υπολογίζει και τυπώνει τα k πλησιέστερα object-ids σε ένα σημείο q=(x,y).

Το πρόγραμμά σας θα πρέπει να παίρνει σαν command-line arguments το αρχείο του R-tree που δημιουργήσατε στο Μέρος 1, το αρχείο των ερωτήσεων NNqueries.txt και τον αριθμό k (αρ. πλησιέστερων γειτόνων). Θα πρέπει να δημιουργεί το δέντρο από τα περιεχόμενα του αρχείου του R-tree και κατόπιν να διαβάζει και να εκτελεί διαδοχικά τις ερωτήσεις από αρχείο ερωτήσεων. Για κάθε ερώτηση πρέπει να τυπώνει στην έξοδο τη γραμμή της ερώτησης (αρχίζοντας από τη γραμμή 0), και τα ids των αποτελεσμάτων, δηλαδή των k πλησιέστερων object MBRs στο σημείο της ερώτησης.

Παράδειγμα (για k=10):

```
0: 9311,7001,803,5361,6764,3905,1642,3260,4669,5762
...
```

Παραδοτέα: Κάντε turnin στο assignment2@mye041 τα προγράμματά σας και ένα PDF αρχείο το οποίο τεκμηριώνει τα προγράμματα και περιέχει διευκρινίσεις ως προς τη λειτουργία τους.

Οδηγίες για τις υποβολές:

1) Μπορείτε να χρησιμοποιήσετε δομές όπως priority queue ή heap από τις βιβλιοθήκες της γλώσσας προγραμματισμού (π.χ. το module heapq της Python) εάν αυτό απαιτείται.

2) Αν χρησιμοποιήσετε Java, το πρόγραμμά σας θα πρέπει να γίνεται compile και να τρέχει και εκτός Eclipse στους υπολογιστές του εργαστηρίου. **Μην χρησιμοποιείτε packages**.

3) Αν χρησιμοποιήσετε Python, μην χρησιμοποιήστε τη βιβλιοθήκη pandas και μην υποβάλετε κώδικα για interactive programming (π.χ. ipython)

4) Υποβάλετε τις εργασίες σας σε ένα **zip** αρχείο (**όχι rar**) το οποίο πρέπει να περιλαμβάνει όλους τους κώδικες καθώς και ένα PDF αρχείο τεκμηρίωσης το οποίο να περιγράφει τη μεθοδολογία σας. **Μην υποβάλετε αρχεία δεδομένων**.

5) Μην ξεχνάτε να βάζετε το όνομά σας (σε greeklish) και το ΑΜ σε κάθε αρχείο που υποβάλετε.

6) Ο έλεγχος των προγραμμάτων σας μπορεί να γίνει σε άλλα αρχεία εισόδου από αυτά που σας δίνονται, άρα θα πρέπει ο κώδικάς σας να μην εξαρτάται από τα συγκεκριμένα αρχεία εισόδου που σας δίνονται.