Διαχείριση σύνθετων δεδομένων

Εργασία 2 - Χωρικά δεδομένα

Βασίλειος Βαλεράς

ΑΜ: 4031

E-mail: cs04031@uoi.gr

Η 2η εργασία είχε ως θέμα την διαχείριση χωρικών δεδομένων. Πιο συγκεκριμένα καλούμαστε να υλοποιήσουμε μια σχάρα (grid) με σκοπό να κάνουμε πιο γρήγορα τα ερωτήματα παραθύρου πάνω στα δεδομένα μας.

Μέρος 1

Για την αναπαράσταση του grid οι δομές που χρησιμοποιήθηκαν είναι οι εξής:

- Ένας πίνακας με τριπλέτες **linestrings** ο οποίος στην 1η θέση έχει το **linestring id**, στην 2η θέση το **minMBR** και **maxMBR** με την μορφή[[MBRminX, MBRminY],[MBRmaxX, MBRmaxY]], και στην 3η θέση τα **points** του linestring με την μορφή [[x1,y1],[x2,y2],…,[xn,yn]].
- Ένας πίνακας **grid**, ο οποίος στην πρώτη θέση έχει το αναγνωριστικό κάθε cell (πχ. [0,0]) και στην δεύτερη θέση έναν πίνακα με τα ελάχιστα/μέγιστα x και y για κάθε cell. Η μορφή του πίνακα grid είναι: [(i,j), [minX, minY, maxX, maxY]].
- Ένα λεξικό **cell_dict** που ως key έχει ένα **αναγνωριστικό cell** και ως value έχει τα **linestrings** που πέφτουν σε κάθε cell.
- Ένα λεξικό **linestring_dict** , που για κάθε cell id ως κλειδί, έχει σαν value το Min-Max MBR του linestring , καθώς και τα points του,

Αρχικά το πρόγραμμα ζητάει από τον χρήστη να του δώσει το αρχείο που θα επεξεργαστούμε. Έπειτα με την βοήθεια της βιβλιοθήκης csv, δημιουργούμε ένα αντικείμενο reader, με το οποίο θα διαβάζουμε κάθε γραμμή του αρχείου. Η πρώτη γραμμή μας υποδεικνύει τον συνολικό αριθμό των linestrings. Έπειτα παρατηρούμε ότι το αρχείο csv αποτελείται από points (x y) χωρισμένα με ‘,’ μεταξύ τους και με κενό ανάμεσα στο x κα το y. Έτσι, σε κάθε γραμμή(linestring) κρατάμε σε μια **temp** μεταβλητή το πρώτο point, και έπειτα το “σπάμε” σε x και y. Αμέσως μετά , για κάθε επόμενο point στην ίδια γραμμή συγκρίνουμε τα x και τα y, ώστε να βρούμε τα min/max MBRs και τα τοποθετούμε σε έναν array **linestring**. Στο τέλος κάθε γραμμής, συγκρίνουμε τα min/max MBRs με τις global μεταβλητές min/max x και y ώστε να βρούμε το ελάχιστο/μέγιστο x y από όλα τα linestrings , πράγμα το οποίο είναι απαραίτητο για την κατασκευή του grid. Τέλος, κάνουμε προσθήκη στον πίνακα **linestrings** το **id**, **MBRs** και **points**.Να σημειωθεί εδώ πως για την εύρεση του id κάθε linestring, έχουμε μια μεταβλητή id η οποία μετά την πρώτη γραμμή του αρχείου αρχικοποιείται σε 1 και την αυξάνουμε κατά 1 κάθε φορά που διαβάζουμε ένα Linestring.

Στην συνέχεια εφόσον έχουμε συλλέξει τα συνολικά min,max x και y από τα linestrings μπορούμε να κατασκευάσουμε το 10x10 grid. Αρχικά βρίσκουμε τα ίσα διαστήματα για τον x και y άξονα (x_interval , y_interval). Στην συνέχεια, για κάθε κελί τα min/max x/y του υπολογίζονται ως εξής: το ελάχιστο x ορίζεται ως το minx του grid + το διάστημα επί την επανάληψη στην οποία βρισκόμαστε(πρακτικά το i), ενώ το μέγιστο x ως το ελάχιστο x του κελιού που μόλις υπολογίσαμε + το διάστημα(το ίδιο και για τον y). Να σημειωθεί επίσης πως ξεκινήσαμε τα iterations στο grid "ανάποδα", με σκοπό η αρχή του grid (0,0) να είναι κάτω αριστερά όπως δείχνει και η εκφώνηση της άσκησης.

Για το cell_dict , αρχικά για κάθε key (cell_id) βάζουμε ως value ένα set() με στόχο να αποφύγουμε τα duplicates, καθώς δεν θέλουμε να έχουμε το ίδιο linestring στο ίδιο cell πάνω από δύο φορές. Στην συνέχεια υλοποιούμε τον αλγόριθμο που περιγράφει η εκφώνηση. Αρχικά κάνουμε initialize δύο μεταβλητές οι οποίες θα κρατάνε για κάθε linestring, το κελί στο οποίο πέφτει το minMBR και το maxMBR του(ώστε μετά να βρούμε και τα ενδιάμεσα κελιά). Έπειτα, για κάθε linestring αλλά και για κάθε cell, αρχικά ελέγχουμε αν το minMBR του Linestring πέφτει μέσα σε κάποιο κελί και αν ναι σε ποιο. Αυτό το κάνουμε τσεκάροντας αν το MBRminX του linestring είναι ανάμεσα από το min και max x του κελιού καθώς και αν το MBRminY του είναι ανάμεσα στα min/max y του κελιού. Αν είναι κρατάμε το id του κελιού και κάνουμε τον ίδιο έλεγχο και για το maxMBR του linestring. Τέλος, για να βρούμε τα ενδιάμεσα κελιά, ελέγχουμε πρώτα αν τα min/max MBRs έπεσαν σε διαφορετικά κελιά. Αν ναι, τότε, παίρνουμε όλα τα κελιά με x και y από το κελί του minMBR έως το maxMBR χωρίς ωστόσο να μετράμε τα ξανά τα cells στα οποία είχαν πέσει αρχικά τα min/max mbrs.

Αμέσως μετά, ακολουθεί η δημιουργία των αρχείων grid.grd και grid.dir, ώστε να τα έχουμε αποθηκευμένα στην μνήμη και έτσι στο μέρος 2 και 3 το grid να φορτώνεται με βάση αυτά. Αρχικά, δημιουργούμε μια βοηθητική δομή – λεξικό **linestring_dict** ώστε να ανακτούμε ένα linestring με όλες του τις πληροφορίες βάζοντας ως key το id του.

Για το αρχείο grid.grd, διατρέχουμε κάθε cell και για κάθε cell , ανακτούμε τα linestrings(ουσιαστικά linestrings ids) που έχει μέσα (**cell_linestrings**).Έπειτα, για κάθε ένα από αυτά τα linestring ids, χρησιμοποιώντας το linestring_dict, στην μεταβλητή ls_info αποθηκεύουμε όλο το Linestring και τέλος το γράφουμε στο αρχείο σύμφωνα με το format της εκφώνησης.

Για το αρχείο grid.dir, αρχικά στην 1η γραμμή γράφουμε τα ελάχιστα/μέγιστα x,y από όλα τα linestrings.Διατηρούμε έναν counter-line ώστε να γράφουμε μπροστά από κάθε line τον αριθμό της γραμμής. Έτσι με ένα for loop, σε κάθε γραμμή γράφουμε το κάθε κελί ταξινομημένο ((0,0) (0,1) κλπ.) και τον αριθμό των linestrings που πέφτουν σε αυτό το κελί(τα ανακτούμε από την δομή cell_dict).

Μέρος 2

Εφόσον έχουμε τα αρχεία grid.grd και grid.dir αποθηκευμένα στην μνήμη μπορούμε να (ξανα)δημιουργήσουμε το grid με βάση αυτά. Η λογική είναι ότι για κάθε cell id που διαβάζουμε στο grid.dir , βλέπουμε πόσα linestrings έχει (έστω x)και στην συνέχεια διαβάζουμε x γραμμές (linestrings) από το αρχείο grid.grd , τα οποία είναι και τα linestrings που πέφτουν στο εκάστοτε cell.

Έτσι, αρχικά αφού κάνουμε open τα αρχεία αποθηκεύουμε την 1η γραμμή του grid.dir που περιέχει τα min/max x/y ώστε να φτιάξουμε τα cells του grid ακριβώς με τον ίδιο τρόπο όπως στο μέρος 1(δημιουργούμε δηλαδή έναν πίνακα grid , που στο 1ο μέρος έχει το αναγνωριστικό του κελιού και στο 2ο τα μέγιστα και ελάχιστα x,y του cell βλ. .γραμμή 51 του κώδικα)

- **linestring_ids:** Ένα set, που χρησιμοποιείται για να αποθηκεύσουμε τα linestring ids.
- **linestrings:** Ο ίδιος πίνακας με το μέρος 1 για την αποθήκευση ενός linestring(id, min-max MBR, points) με την διαφορά ότι δεν έχουμε duplicates, καθώς τα φιλτράραμε με την βοήθεια του linestring_ids.Αυτό το κάνουμε γιατί όπως θα δούμε στην συνέχεια δεν θέλουμε να έχουμε το ίδιο Linestring παραπάνω από μια φορά στο ίδιο κελί.
- **cell_dict:** Ίδια δομή με το μέρος 1.

Έπειτα, για κάθε γραμμή του grid.dir , κάνουμε split με το κενό και έτσι δημιουργείται ένας πίνακας parts. Το cell id είναι το (parts[1],. parts[2]) και ο αριθμός των Linestrings στο κελί το parts[3]. Αν, τα linestrings δεν είναι 0 , διαβάζουμε από το αρχείο grid.grd όσα linestrings (ουσιαστικά Lines) έπεσαν στο κελί, κάνοντάς τες split με ‘,’' (πίνακας linestring_data). Έπειτα, για κάθε linestring σε αυτόν τον πίνακα ανακτούμε το linestring id , και αν αυτό δεν είναι στο σετ που αποθηκεύουμε τα linestrings, τότε το τοποθετούμε. Αμέσως μετά αποθηκεύουμε σε temp μεταβλητές τα min/max MBRs και τα points του linestring, και στην συνέχεια τα κάνουμε append στον πίνακα linestrings, δημιουργώντας έτσι την δομή που είχαμε στο μέρος 1. Τέλος, συμπληρώνουμε με βάση το cell id και τα linestring ids το cell_dict που είχαμε στο μέρος 1. Να σημειωθεί πώς αν τα linestrings στο αρχείο dir είναι 0 (σε μια γραμμή) βάζουμε στο cell_dict έναν κενό πίνακα σαν value.

Για τις ερωτήσεις παραθύρου, αρχικά διαβάζουμε το queries.txt. Για κάθε line(ερώτηση), την κάνουμε split με ‘,’ και δημιουργούμε το window (window[0] : minX, window[1] : maxX, window[2] : minY, window[3] : maxY) .Έπειτα, κάνουμε τον έλεγχο για το ποια κελιά από το grid έχουν επικάλυψη με το παράθυρο. Επικάλυψη έχουμε όταν το minX του window είναι μικρότερο ή ίσο από το maxX του κελιού ***και*** όταν το maxX του window είναι μεγαλύτερο ή ίσο από το minX του κελιού. Παρομοίως, πρέπει ***και*** στον άξονα y να υπάρχει επικάλυψη άρα: όταν το minY του window είναι μικρότερο ή ίσο από το maxY του κελιού ***και*** όταν το maxY του window είναι μεγαλύτερο ή ίσο από το minY του κελιού. Έτσι, αν το κελί έχει επικάλυψη με το παράθυρο, κάνουμε appends στον πίνακα records τα linestrings του εκάστοτε κελιού (δεν έχουμε duplicates στο ίδιο κελί) καθώς και το ίδιο το cell για χρήση αργότερα. Αυξάνουμε επίσης τον μετρητή intersect_cells , ώστε να τον τυπώσουμε στο output. Έπειτα για κάθε record (linestrings από cells που έκαναν intersect με το παράθυρο) και cell κάνουμε τους εξής ελέγχους: Για κάθε linestring στο record, και για όλα τα Linestrings στον πίνακα linestrings, όταν βρεθεί linestring που να είναι ίσο με το

Linestring του record, βρίσκουμε το reference point σύμφωνα με την εκφώνηση και ελέγχουμε αν το MBR του συγκεκριμένου linestring έχει επικάλυψη με το MBR του παραθύρου. Ο τρόπος να ελέγξουμε την επικάλυψη είναι ακριβώς ο ίδιος που περιγράψαμε παραπάνω για την επικάλυψη cell-window. Τελευταίος έλεγχος είναι αν το reference point του linestring είναι μέσα στο συγκεκριμένο cell που εξετάζουμε, και αν ναι, τότε προσθέτουμε το id του στα outputs.

Μέρος 3

Το μέρος 3 ουσιαστικά αποτελεί βελτίωση του μέρους 2 και εργαζόμαστε πάνω στον πίνακα outputs, που περιέχει τα linestrings των οποίων το MBR έχει επικάλυψη με το παράθυρο. Έτσι για κάθε τέτοιο linestring, αρχικά ελέγχουμε αν το MBR του έχει ολική επικάλυψη σε έναν από τους άξονες x ή y. Δηλαδή ελέγχουμε αν τον MBRminX **και** MBRmaxX του linestring είναι ανάμεσα από το max και min x του παραθύρου **ή** το MBRminY **και** MBRmaxY του linestring είναι ανάμεσα από το max και min y του παραθύρου. Αν ναι , κάνουμε append τα συγκεκριμένα linestring στο output. Αν όχι τότε πρέπει να τσεκάρουμε ξεχωριστά κάθε line segment του linestring (ουσιαστικά line segment είναι το ευθύγραμμο τμήμα που ενώνει 2 συνεχόμενα points x1,y1 & x2,y2 του linestring) αν κάνει intersect με κάθε πλευρά από το window(line segment του window). Έτσι, με ένα for loop εφόσον ανακτήσουμε τα x1,y1 & x2,y2 (δηλαδή όλα τα συνεχόμενα line segments) αλλά και τα wx1,wx2 wy1,wy2(πχ. για wx1 = window minX , wx2 = window maxX,wy1 = window minY,wy2 = window minY θα δούμε αν το linestring segment κάνει intersect μες την κάτω πλευρά του window) που είναι τα points που σχηματίζουν το παράθυρο(window line segments) καλούμε την συνάρτηση check_intersection, οποία μας επιστρέφει true αν κάνουν intersect.

Οδηγίες χρήσης προγράμματος

Αρχικά έχετε στον ίδιο κατάλογο με τα προγράμματα το αρχείο .csv με τα Linestrings και το αρχείο .txt με τα queries.

Έπειτα στο terminal για το 1^{ο} μέρος τρέχετε: python meros1.py . Μετά έπειτα από σχετικό μήνυμα του προγράμματος θα πρέπει να πληκτρολογήσετε το αρχείο .csv .

Για το μερος 2 και 3 : python meros2.py ή python meros3.py (αφού έχετε τρέξει φυσικά το μέρος 1) και έπειτα μετά από το μήνυμα που θα δείτε στην κονσόλα θα πρέπει να πληκτρολογήσετε το όνομα του αρχείου .txt με τα queries.