

Η άσκηση υλοποιήθηκε σε γλώσσα προγραμματισμού Java και αποτελείται από 5 διαφορετικές κλάσεις. Την κλάση Record για την αναπαράσταση των αντικειμένων στο αρχείο tiger_roads.csv, την κλάση Cell για το κάθε κελί του Grid, την κλάση GridCreator για την δημιουργία του Grid, την κλάση SelectionQuery που αποτιμά τα ερωτήματα από το queries.txt και την RefinementStep όπου επεκτείνει την προηγούμενη κλάση κάνοντας και refinement.

Η κλάση Record.java περιέχει τα 3 πεδία ID, MBR, linestring.

Η κλάση Cell.java ορίζει τα όρια του κάθε Cell(xmin, xmax, ymin, ymax) και έχει δύο μεθόδους **intersects** που χρησιμοποιούνται για τον έλεγχο αν υπάρχει τομή μεταξύ του Cell και ενός MBR και του Cell με ένα query.

Στην κλάση GridCreator.java ουσιαστικά υλοποιείται το 1^ο μέρος της άσκησης και η δημιουργία του grid. Με την κλήση της main της κλάσης αυτής καλούνται οι μέθοδοι που είναι υπεύθυνες για την δημιουργία των αρχείων grid.dir και grid.grd. Αρχικά, φορτώνονται τα αντικείμενα(records) από το csv αρχείο μέσω της μεθόδου **createRecords()** όπου η μέθοδος αυτή αφού καλέσει αρχικά την μέθοδο **readFile()** για να πάρει τις γραμμές του αρχείου, εν συνεχεία κατασκευάζει τα records και βρίσκει τις ελάχιστες και μέγιστες τιμές που θα έχει το grid. Μετά, δημιουργείται το grid μέσω της μεθόδου **createGrid()** όπου αφού βρω αρχικά τα xInterval και yInterval, δημιουργώ ένα ένα τα Cell και τα προσθέτω στον πίνακα που κρατά αυτά, το grid. Στην συνέχεια, για κάθε MBR των αντικειμένων θα πρέπει να βρούμε ποια κελιά της σχάρας τέμνει καλώντας την μέθοδο **findIntersections()**. Η μέθοδος αυτή επιστρέφει ένα HashMap cellContents με κλειδί την συντεταγμένη του cell και τιμές τα ids των records που τέμνει. Τέλος, καλούνται οι μέθοδοι **writeGridGrd()** και **writeGridDir()** όπου χρησιμοποιώντας το cellContents δημιουργούν τα αντίστοιχα αρχεία.

Στην κλάση Selection Query.java υλοποιείται το 2^ο μέρος της άσκησης για την αποτίμηση ερωτήσεων που βρίσκονται στο αρχείο queries.txt με χρήση reference point για duplicate detection. Η κλάση αυτή έχει ως πεδίο το grid και το HashMap cellContents, τα οποία γεμίζουν με την κλήση της μεθόδου **loadGrid()**. Η loadGrid() διαβάζει αρχικά την πρώτη γραμμή από το grid.dir ώστε να φτιάξει το grid και παίρνοντας από κάθε επόμενη γραμμή του αρχείου το cell και τον αριθμό των records που τέμνονται με αυτό, στην συνέχεια διαβάζει ανά τον αριθμό των records(3^η στήλη του grid.dir) από το grid.grd και γεμίζει το cellContents. Ύστερα, με την μέθοδο **readQueries()** διαβάζουμε το αρχείο queries.txt και η μέθοδος επιστρέφει τις ερωτήσεις σε μορφή λίστας με μορφή xmin, xmax, ymin, ymax για

την κάθε μια ερώτηση. Για την αποτίμηση των ερωτήσεων καλείται η μέθοδος **answerQueries()**, η οποία αφού καλέσει αρχικά την **readQueries()** για να πάρει τις ερωτήσεις, στην συνέχεια για κάθε ερώτηση χρησιμοποιώντας το **cellContents** ελέγχει αν κάποιο cell τέμνεται με την ερώτηση(query). Αν τέμνεται ,τότε ελέγχεται αν κάποιο MBR που περιέχεται στο cell τέμνεται όντως με την ερώτηση. Έτσι, τέλος αν τέμνεται και το reference point του και αυτό περιέχεται στο κελί που εξετάζεται, τότε θα προστεθεί το id του στο πίνακα με τα ids που τέμνονται με την ερώτηση. Τέλος, τυπώνουμε τα αποτελέσματα στο console όπως βρίσκονται και στο αρχείο results.

```
Console [x]
<terminated> SelectionQuery [Java Application] C:\Program Files\Java\jdk-12.0.2\bin\javaw.exe (21 Απρ 2023, 2:44:36 μ.μ.)
Meros 2

Query 1 results:
[13151, 15262, 15774, 16782, 21379, 22260, 22500, 22946, 22947]
Cells: 1
Results: 9
-----
Query 2 results:
[33887, 34512, 34862]
Cells: 1
Results: 3
-----
Query 3 results:
[30397]
Cells: 2
Results: 1
-----
Query 4 results:
[1108]
Cells: 1
Results: 1
-----
Query 5 results:
[5496]
Cells: 3
Results: 1
-----
```

Στην κλάση **RefinementStep.java** υλοποιείται το 3^ο μέρος της άσκησης, δηλαδή το refinement stage. Προστέθηκε η μέθοδος **intersects(List<List<Double>> lineSegment, List<Double> query)**, η οποία ελέγχει αν ένα line segment από το linestring τέμνεται με κάποιο ευθύγραμμο τμήμα της ερώτησης σύμφωνα με τους τύπους για την εύρεση τομών μεταξύ δύο ευθειών που δίνονται. Έτσι, η **answerQueries()** επεκτάθηκε ώστε να εντοπίζει τα MBR που όντως τέμνονται με ένα query, βρίσκοντας αρχικά αν το MBR του υπερκαλύπτεται εντελώς σε τουλάχιστον μία από τις δύο διαστάσεις x,y από το παράθυρο επιλογής και αν αυτό δεν συμβαίνει τότε ελέγχουμε τα μεμονωμένα ευθύγραμμα τμήματα του linestring και εξετάζουμε αν κάποιο τέμνει τις πλευρές του παραθύρου με χρήση της **intersects()** μεθόδου.

Τέλος, τυπώνουμε τα αποτελέσματα:

Console

<terminated> RefinementStep [Java Application] C:\Program Files\Java\jdk-12.0.2\bin\javaw.exe (21 Απρ 2023, 4:35:41 μ.μ.)

Meros 3

Query 1 results: |
[21379, 22260, 22500, 22946, 22947]

Cells: 1
Results: 5

Query 2 results:
[34512]

Cells: 1
Results: 1

Query 3 results:
[30397]

Cells: 2
Results: 1

Query 4 results:
[]

Cells: 1
Results: 0

Query 5 results:
[5496]

Cells: 3
Results: 1
