

Διαχείριση Σύνθετων Δεδομένων

2^η Σειρά Ασκήσεων

Βασιλειάδης Μιλτιάδης 2944

Ερώτημα πρώτο

Τρέχετε το πρόγραμμα εκτελώντας

```
1. python <rtree.py> <coordsfile> <offsetsfile>
```

Αρχικά διαβάζεται το αρχείο με τα offsets από το αρχείο αυτό το πρόγραμμα τραβάει τις γραμμές από το αρχείο με τις συντεταγμένες και το id αυτά μπαίνουν σε μια λίστα που ονομάζεται polygon. Από το polygon βρίσκω τα min και max των x y με τις μεθόδους map min max και lambda algebra για να σχηματιστεί το mbr του πολυγώνου. Το mbr προστίθεται με το id του στην λίστα mbrs Βρίσκω το κέντρο του πολυγώνου που είναι ο μεσος όρος των συνιστωσών στους άξονες x και y καλώ την συνάρτηση για το z curve και τα προσθέτω σε ένα άλλο πίνακα μαζί με το id του πολυγώνου. Sortαρω αυτόν τον πίνακα και μετά κανω τον πίνακα των mbrs reorder βάση την σειρά των ids στο σαρταρισμένο πίνακα με τα zcurves. Εδώ με βόλεψε ότι τα πολύγωνα ήταν zero indexed αλλιώς θα έπρεπε να τα προσθεσω όλα στον αρχικό πίνακα και να κανω ταξινόμηση με κλειδί τα zcurves.

Για την δημιουργία του δέντρου χρειάζονται να κρατάω 2 λίστες η μια λίστα Tree περιέχει το ίδιο το δέντρο και την queue η οποία είναι μια απλή FIFO ουρά προτεραιότητας. Για να πακεταριστούν τα MBRs σε φύλλα παίρνω τα ταξινομημένα πλέον mbr από την λίστα ανά 20 και τα βάζω σε μια λίστα με αυξάνοντα node_ids και το επίπεδο που βρίσκονται. Και μπαίνουν και στο Tree με την σήμανση ότι πρόκειται για φύλλο και στην ουρά.

Για την δημιουργία των nodes που δεν είναι φύλλα ακολουθώ τον εξής αλγόριθμο

Αν ουρά έχει πάνω από 1 στοιχείο:

Όσο το επίπεδο στο πανω πανω στοιχείο της ουρας είναι το τρέχον.

Απωθώ από την ουρά στοιχεία μέχρι να φτασω τα 20 ή μεχρι να αλλαξει το επίπεδο

Φτιάχνω ένα νέο node με $id = id + 1$ και προσθέτω στο node στην ουρά με $lvl = lvl$ (τρέχον + 1) και στον πίνακα Tree με την σήμανση ότι πρόκειται για node.

Αυτό εκτελείται αναδρομικά μέχρις ότου να απομείνει ένα node στην ουρά. Το node αυτό θα είναι η ρίζα οπότε μπαίνει κατ ευθείαν στον Tree.

Για να έχω σχετικά καλό occurance στους κόμβους κοιτάω για τους κόμβους στο ίδιο επίπεδο αν ο τελευταίος που δημιουργηθηκε έχει μηκος κατω από 8 αν ναι κλέβω την διαφορά από τον αμεσως προηγουμενο του κόμβο με slicing και το τοποθετώ στην αρχή του τελευταίου κόμβου για να διατηρηθει το zorder

Για να τυπωθεί η λίστα στο αρχείο όπως ζητήθηκε στην εκφώνηση (δηλαδή να διαβάζεται ως λίστες)

Μετατρέπω κάθε γραμμή της λίστας σε string και την γράφω στο αρχείο.

Ερώτημα δεύτερο

Τρέχετε το πρόγραμμα εκτελώντας

```
1. python <rangequeries.py> <rtreefile> <queriesfile>
```

Για να γίνει κατανοητή από την python η μορφή της λίστας όπως διαβάζεται από το αρχείο, έκανα χρήση της `ast.literal_eval(line)` για κάθε γραμμή του αρχείου

Για το δεύτερο ερώτημα ξεκινά η αναζήτηση από την ρίζα του δέντρου αν το mbr στο κάθε node ικανοποιεί αυτή την ιδιότητα το ένα παράθυρο να μην είναι ούτε δεξιά ούτε αριστερά ούτε πάνω ούτε κάτω από το άλλο και αυτό γίνεται με τις συγκρίσεις παρακάτω, όπου `q_xxxx` εννοείται το query

```
1. not (x_high <= qx_low or x_low >= qx_high or y_high <= qy_low or y_low >= qy_high
)
2.
```

Τότε το εν λόγω mbr τέμνει το παράθυρο του query η εκτέλεση συνεχίζεται ακολουθώντας τα nodes που τέμνουν μέχρι να φτάσει στα φύλλα του δέντρου όπου για κάθε καταχώρηση στο φύλλο που τέμνει το παράθυρο περνά τα ids στο αποτέλεσμα και τα τυπώνει μαζί με το πλήθος τους για κάθε query που διαβάζεται από το αρχείο.

Ερώτημα τρίτο

Τρέχετε το πρόγραμμα εκτελώντας

```
1. python <bestnn.py> <rtreefile> <queriesfile>
```

Για να γίνει κατανοητή από την python η μορφή της λίστας όπως διαβάζεται από το αρχείο, έκανα χρήση της `ast.literal_eval(line)` για κάθε γραμμή του αρχείου

Για την υλοποίηση του ερωτήματος αυτού κατανόησα την έννοια της απόστασης σημείου από mbr ως την ρίζα του τετραγώνου των διαφορών μεταξύ των x και των y ανάλογα την θέση του αντικειμένου (αριστερά δεξιά πάνω ή κάτω)

```
1.     if x < mxmin:
2.         dx = mxmin - x
3.     if x > mxmax:
4.         dx = x - mxmax
5.     if y < mymin:
6.         dy = mymin - y
7.     if y > mymax:
8.         dy = y - mymax
9.     return sqrt( dx**2 + dy**2 )
10.
```

Στην υλοποίηση του best η δεν έχω καταλάβει σε ποιο σημείο απωθώ το object από την Q για να το τοποθετήσω στο αποτέλεσμα. Η υλοποίηση είναι όπως περιγράφεται σχεδόν στις σημειώσεις του μαθήματος, μου φαίνεται ότι θα μπορούσα να βρω την λύση δυστυχώς εξέπνευσε η προθεσμία για την άσκηση.