ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Δομές Δεδομένων και Αλγόριθμοι

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Άσκηση 2 - Δεικτοδότηση Διανυσματικών Δεδομένων

18.05.2023

1: Περιγραφή Κώδικα

1.1: Ανάλυση Πακέτων

Ο κώδικας χωρίζεται σε 4 πακέτα:

1. model: Περιέχει τις υλοποιήσεις για τα δέντρα kd-Tree και PR-QuadTree. Για να υλοποιηθούν οι παραπάνω δομές δεδομένων, έχουν δημιουργηθεί οι κλάσεις Data, Node, PRData και PRNode.
   - Η κλάση Data είναι μια generic κλάση η οποία αντιπροσωπεύει ένα δισδιάστατο σημείο του kd-Tree, και περιέχει τα πεδία x, y (2 generic comparables), το value (μια generic μεταβλητή η οποία είναι συσχετισμένη με το σημείο (x,y) στο kd-Tree), και το πεδίο depth που χρησιμοποιείται για να γίνει σύγκριση ανάμεσα στα (x,y) στο kd-Tree (αν το βάθος είναι άρτιος αριθμός γίνεται σύγκριση με βάση το x, αλλιώς γίνεται σύγκριση με βάση το y).
   - Η κλάση Node αντιπροσωπεύει έναν κόμβο στο kd-Tree. Περιέχει τα πεδία data (τα δεδομένα του κόμβου), left (το αριστερό παιδί του κόμβου) και right (το δεξιό παιδί του κόμβου).
   - Η κλάση PRData είναι παρόμοια με την κλάση Data, με τη διαφορά ότι τα x και y σε αυτή την περίπτωση δεν είναι generic μεταβλητές.
   - Η κλάση PRNode αντιπροσωπεύει έναν κόμβο του PR-QuadTree και περιέχει τις μεθόδους εισαγωγής και αναζήτησης στο δέντρο.
2. modelTesting: Περιέχει τις εξής κλάσεις:
   - DataPool : δεξαμενή δεδομένων στην οποία αποθηκεύονται N τυχαία σημεία (x,y) τα οποία εισάγονται στα δέντρα.
   - TestGenerator : κλάση - thread η οποία εισάγει M (50 - 100000) δεδομένα στο kd-Tree και PR-QuadTree, κάνει 100 επιτυχημένες και αποτυχημένες αναζητήσεις σε αυτά και εξάγει τα αποτελέσματά τους σε μια λίστα.
   - App : Περιέχει τη main() η οποία τυπώνει τα αποτελέσματα από το TestGenerator στην κονσόλα.
   - TestStructure : κλάση αποθήκευσης μετρήσεων.
3. unitTesting: Περιέχει μερικά unit tests για να διαπιστωθεί η σωστή λειτουργία των αλγορίθμων που υλοποιήθηκαν.
4. utils: Περιέχει τις κλάσεις Utils, CsvWriter. Η Utils έχει μερικές χρήσιμες static μεθόδους, και η κλάση CsvWriter γράφει τα αποτελέσματα των μετρήσεων σε ένα αρχείο csv.

1.2: Εισαγωγή στο kd-Tree

Η εισαγωγή ενός νέου σημείου στο kd-Tree υλοποιήθηκε ως εξής:
Αν η ρίζα του δέντρου είναι κενή, το σημείο εισάγεται στη ρίζα του δέντρου. Ειδάλλως, εισάγεται στο δεξί ή αριστερό υποδέντρο, αν το σημείο είναι μεγαλύτερο ή μικρότερο, αντίστοιχα. Η σύγκριση γίνεται με βάση το βάθος στο οποίο βρίσκεται το σημείο στο δέντρο (κάθε φορά που πηγαίνουμε αριστερά ή δεξιά το βάθος του σημείου αυξάνεται κατά 1).

1.3: Αναζήτηση στο kd-Tree

Η αναζήτηση ενός σημείου στο kd-Tree υλοποιήθηκε ως εξής: Αν το σημείο που αναζητούμε είναι μεγαλύτερο ή ίσο με το σημείο του κόμβου τότε αυξάνουμε το βάθος του σημείου κατά 1 και αναζητούμε στο δεξί υποδέντρο, ειδάλλως αναζητούμε στο αριστερό υποδέντρο. Επαναλαμβάνουμε τα βήματα αναδρομικά έως ότου να βρεθεί το σημείο ή μέχρι να καταλήξουμε σε κενό κόμβο.

1.4: Εισαγωγή στο PR-QuadTree

Η εισαγωγή ενός σημείου στο PR-QuadTree υλοποιήθηκε ως εξής:
Αν ο κόμβος δεν περιέχει δεδομένα και είναι φύλλο τότε εισάγουμε το σημείο στον κόμβο. Ειδάλλως, αν είναι ενδιάμεσος κόμβος, χωρίζουμε τον κόμβο σε 4 υποπεριοχές (NW, NE, SW, SE) και εισάγουμε το σημείο σε μια από αυτές. Τέλος, αν ο κόμβος περιέχει δεδομένα, αλλά δεν είναι φύλλο, εισάγουμε εκ νέου έτσι ώστε μόνο τα φύλλα να περιέχουν δεδομένα.

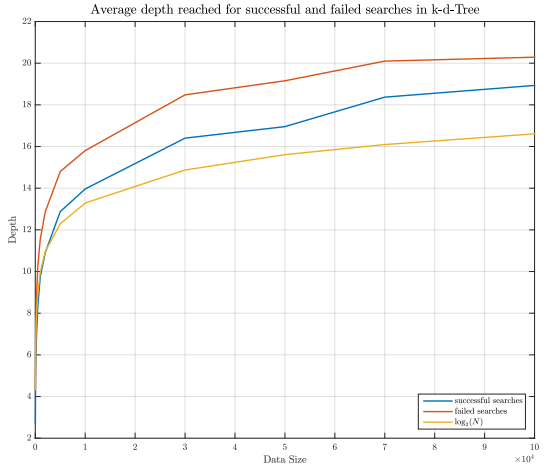
1.5: Αναζήτηση στο PR-QuadTree

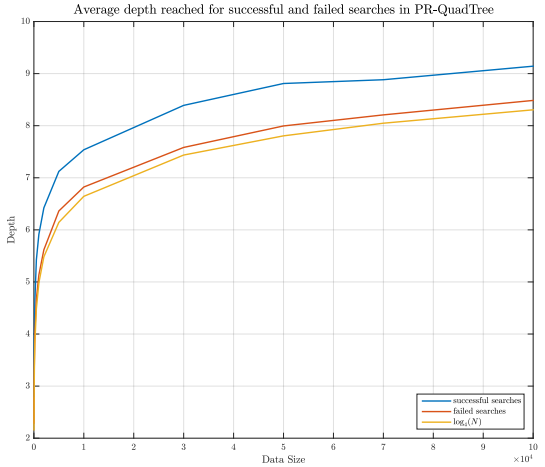
Η αναζήτηση ενός σημείου στο PR-QuadTree υλοποιήθηκε ως εξής:
Όσο βρισκόμαστε σε ενδιάμεσο κόμβο διασχίζουμε το δέντρο, και κάθε φορά πηγαίνουμε σε μία από τις 4 υποπεριοχές (NW, NE, SW, SE) συγκρίνοντας κάθε φορά το σημείο με τα όρια της κάθε υποπεριοχής, έως ότου καταλήξουμε σε φύλλο.

2: Σύγκριση μεθόδων και μετρήσεις

Τα αποτελέσματα των μετρήσεων φαίνονται στον παρακάτω πίνακα:

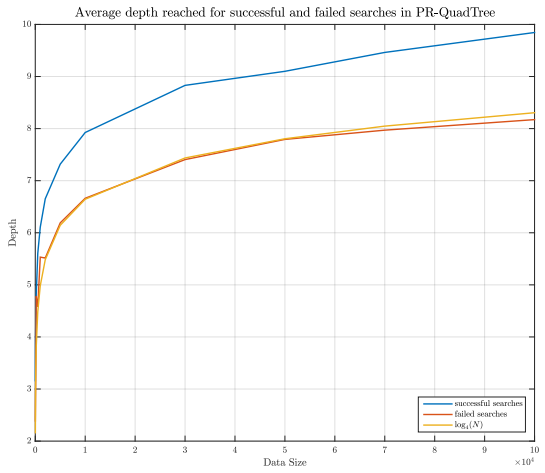
Μέσος όρος βάθους δέντρου				
Data Size	Successful Searches in kd-Tree	Failed searches in kd-Tree	Successful Searches in PR-QuadTree	Failed searches in PR-QuadTree
50	5.67	7.61	3.78	3.08
100	7.17	8.94	4.33	3.49
200	8.33	10.07	4.72	4.07
500	9.86	12.17	5.35	4.62
1000	11.23	13.41	5.97	5.23
2000	12.64	14.47	6.38	5.73
5000	14.47	16.89	7.06	6.30
10000	15.92	18.35	7.57	6.80
30000	18.34	20.27	8.32	7.67
50000	19.03	21.56	8.79	8.03
70000	19.74	22.00	8.94	8.25
100000	20.59	22.77	9.16	8.51



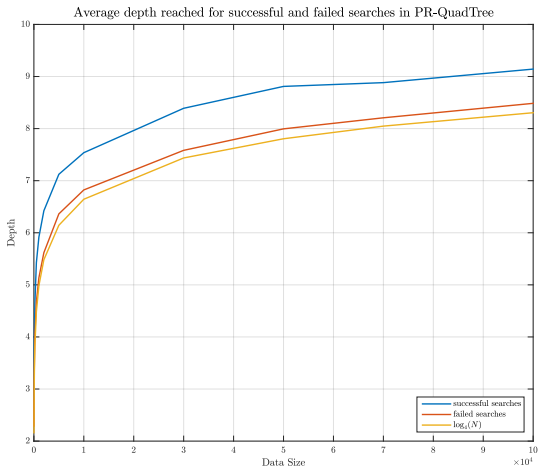


3: Ανάλυση Αποτελεσμάτων

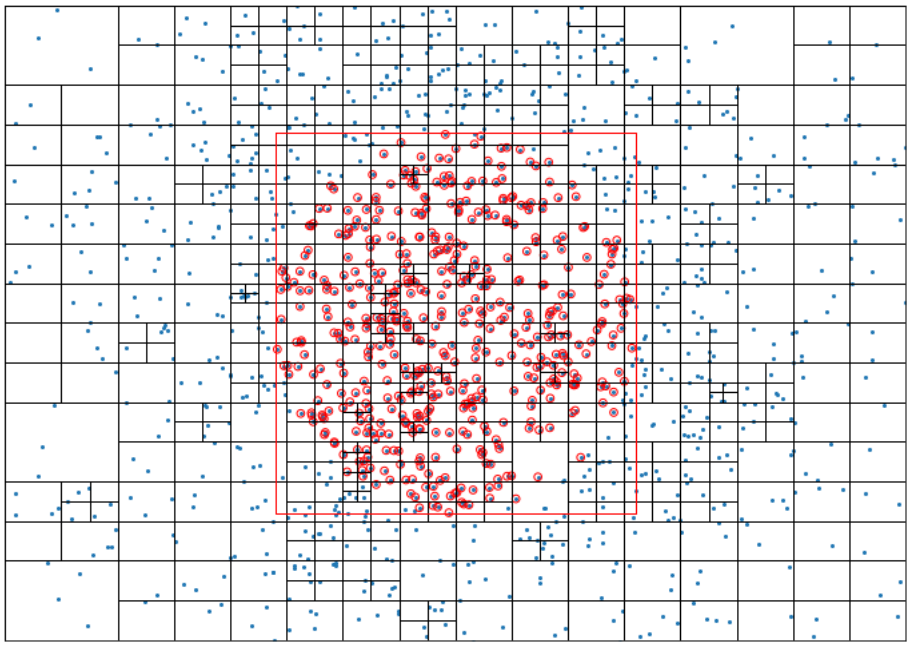
Μπορούμε να παρατηρήσουμε ότι και στις δύο περιπτώσεις η πολυπλοκότητα για την αναζήτηση τυχαίων κλειδιών στο kd-Tree και στο PR-QuadTree μεγέθους N, είναι λογαριθμική. Επίσης, είναι εμφανές ότι το μέσο βάθος στο PR-QuadTree είναι περίπου το μισό σε σύγκριση με το μέσο βάθος στο kd-Tree, το οποίο είναι αναμενόμενο αφού το PR-QuadTree έχει διπλάσιους κόμβους. Μια ενδιαφέρουσα διαφοροποίηση μεταξύ των kd-Tree και PR-QuadTree είναι ότι το μέσο βάθος για την αναζήτηση δεδομένων που δεν υπάρχουν στο δέντρο είναι μικρότερο συγκριτικά με το μέσο βάθος αναζήτησης δεδομένων που υπάρχουν στο δέντρο. Μια πιθανή εξήγηση για αυτό το φαινόμενο είναι ότι στο PR-QuadTree οι περιοχές δεν ομοιόμορφα πυκνές, με αποτέλεσμα το δέντρο να μην είναι ισόρροπο και αν γίνουν αναζητήσεις δεδομένων που δεν υπάρχουν στο δέντρο, το βάθος στο οποίο τερματίζει η αναζήτηση είναι μικρότερο σε σχέση με το ύψος του δέντρου. Αυτό είναι ιδιαίτερα εμφανές αν αντί για ομοιόμορφα κατανεμημένα τυχαία σημεία επιλέξουμε να εισάγουμε στο PR-QuadTree δεδομένα που ακολουθούν κανονική κατανομή με μέση τιμή N/2 και διασπορά 1. Οι διαφοροποιήσεις γίνονται μεγαλύτερες όπως μπορούμε να δούμε και παρακάτω:



(a) Gaussian data



(b) Uniform data



Εικόνα 3: gaussian data visualized

4: Οδηγίες εκτέλεσης προγράμματος

Για την εκτέλεση του προγράμματος είναι απαραίτητη η έκδοση 14 του Java Development Kit καθώς στο πρόγραμμα χρησιμοποιούνται records.

Πηγές

1. https://scipython.com/blog/quadtrees-2-implementation-in-python/
2. https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/PRquadtree.html
3. https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/KDtree.html
4. σημειώσεις φροντιστηρίου του μαθήματος