ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΩΝ

Αναφορά 2ης εργαστηριακής άσκησης

Χαϊντούτη Μαρία || ΑΜ 2020030129

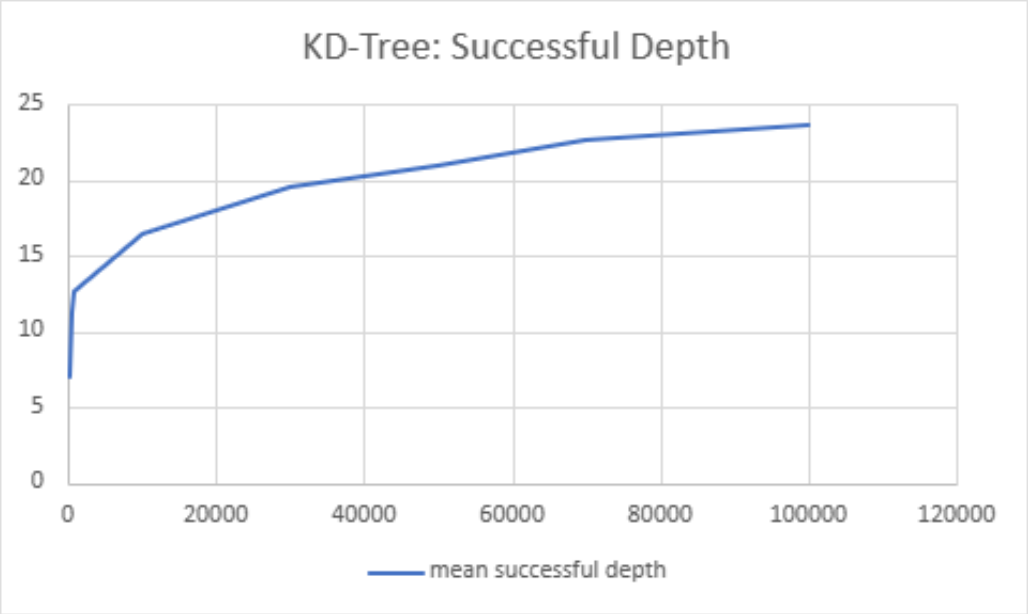
Η main κλάση βρίσκεται στο πακέτο Main. Το πρόγραμμα μεταγλωττίζεται επιτυχώς, υλοποιώντας και τις δύο ζητούμενες δομές KD-Tree και PR Quad-Tree χωρίς χρονικές καθυστερήσεις. Για την υλοποίηση της άσκησης χρησιμοποιήθηκαν οι ακόλουθες εξωτερικές πηγές πληροφόρησης:

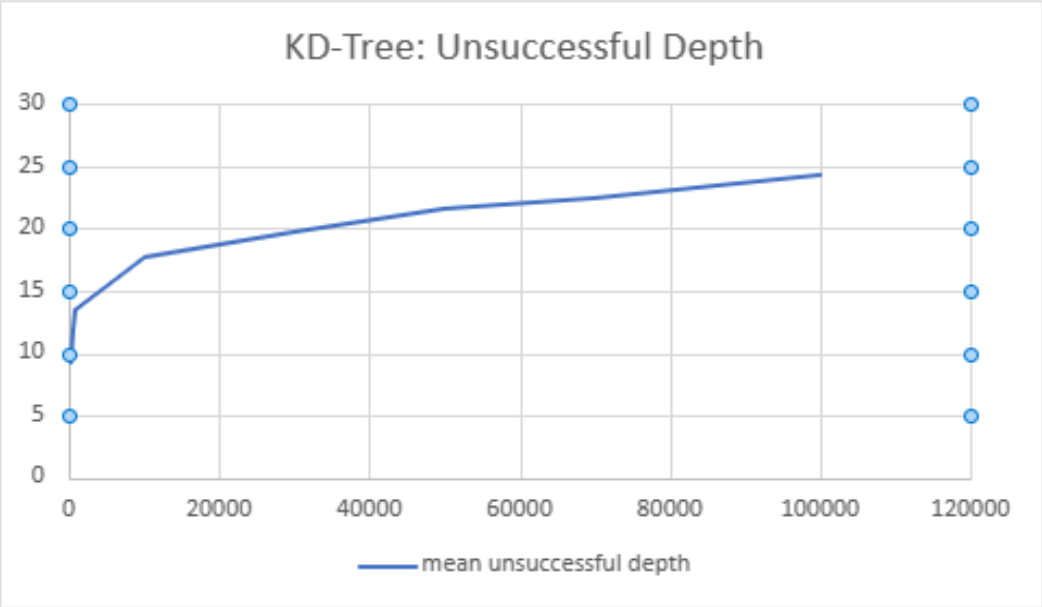
- Search and Insertion in K Dimensional tree - GeeksforGeeks
- Binary Search Tree In Java - Implementation & Code Examples (softwaretestinghelp.com)
- https://github.com/panas007/Pr-QuadTree/blob/master/Qtree.java
- Quadtree - Wikipedia
- k-d tree - Wikipedia

Γενική Ιδέα KD-Tree:

Η βασική ιδέα ενός KD - Tree είναι να διαχωρίζει τον χώρο στις διαστάσεις των σημείων κατά εναλλασσόμενο τρόπο. Σε κάθε επίπεδο του δέντρου, επιλέγεται μια διάσταση ,στην προκειμένη περίπτωση ή x ή y ,για το διαχωρισμό και τα σημεία τοποθετούνται στον κόμβο ανάλογα με την τιμή της επιλεγμένης διάστασης. Στον παραδοτέο κώδικα η κλάση KDTree αντιπροσωπεύει ένα KD-Tree με δυνατότητες εισαγωγής (insert) και αναζήτησης (KDsearch) σημείων 2D. Ο κόμβος του δέντρου (KDNode) περιέχει ένα σημείο (KDPoint) καθώς και αναφορές στο αριστερό και δεξί παιδί του. Ο κώδικας υλοποιεί την εισαγωγή σημείων στο δέντρο με βάση τις τιμές των συντεταγμένων των σημείων. Η αναζήτηση ενός σημείου στο δέντρο γίνεται ακολουθώντας το μοτίβο του διαχωρισμού του χώρου , που περιγράφεται παραπάνω. Ο κώδικας υπολογίζει το μέσο βάθος επιτυχημένων και ανεπιτυχών αναζητήσεων για ένα σύνολο τυχαίων σημείων που εισάγονται στο δέντρο, καθώς και τον χρόνο εκτέλεσης των αναζητήσεων.

- Τα διαγράμματα, τα οποία δείχνουν πώς μεταβάλλεται το βάθος στις πετυχημένες και μη αναζητήσεις είναι τα παρακάτω:



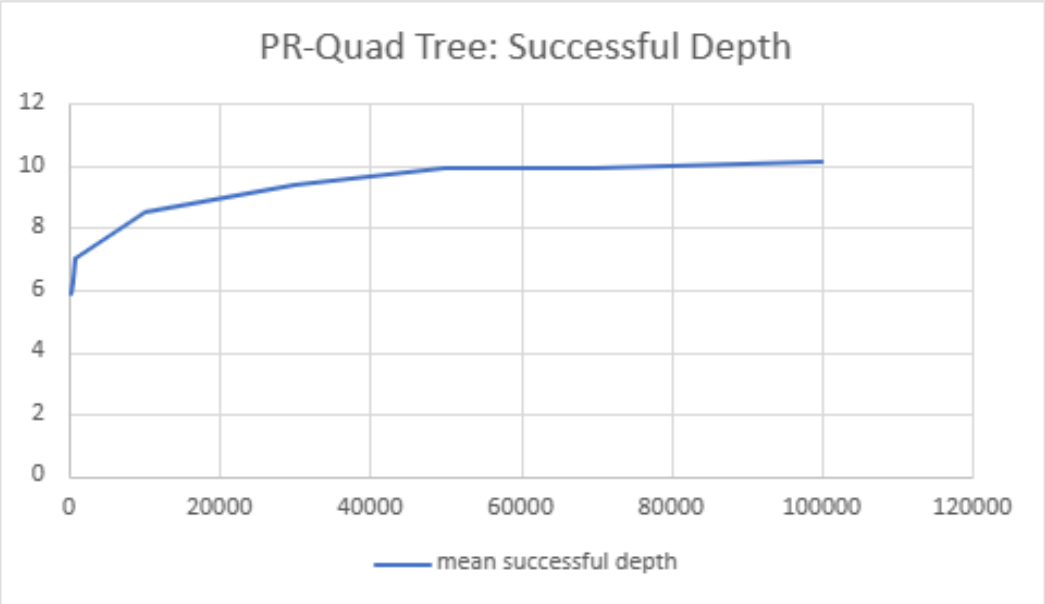


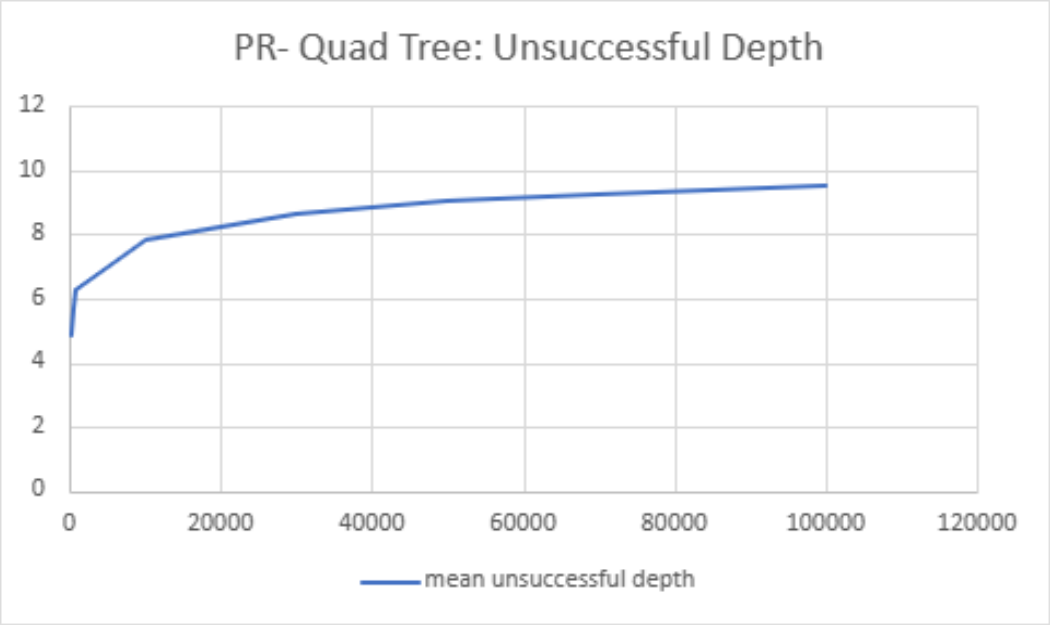
- Η πολυπλοκότητα που χαρακτηρίζει τις παραπάνω καμπύλες είναι η Olog(N), όπου N ο αριθμός των στοιχείων που έχουν εισαχθεί στο δέντρο.

Γενική Ιδέα PR Quad-Tree:

Η γενική ιδέα ενός Quad-Tree είναι να διασπάσει αναδρομικά τον χώρο σε τέσσερα ίσα τμήματα και να αποθηκεύσει δεδομένα σε κατάλληλους κόμβους. Αυτό επιτρέπει αποδοτική αναζήτηση και επεξεργασία των δεδομένων που υπάρχουν σε έναν συγκεκριμένο τμήμα του χώρου. Ο παραδοτέος κώδικας περιλαμβάνει τις κλάσεις QuadTree και QuadNode, οι οποίες αντιπροσωπεύουν τον Quad-Tree και τους κόμβους του αντίστοιχα. Η Quad-Tree κλάση χρησιμοποιείται για την εισαγωγή σημείων και την αναζήτησή τους βάσει των συντεταγμένων τους. Επίσης, ο κώδικας περιλαμβάνει μια κύριο μέθοδο main που εισάγει τυχαία σημεία στο Quad-Tree και αναζητά τυχαία σημεία μέσα σε αυτό. Τέλος, υπολογίζει τον μέσο χρόνο αναζήτησης και τον μέσο βάθος για επιτυχημένες και μη επιτυχημένες αναζητήσεις.

- Τα διαγράμματα, τα οποία δείχνουν πώς μεταβάλλεται το βάθος στις πετυχημένες και μη αναζητήσεις είναι τα παρακάτω:





- Η πολυπλοκότητα που χαρακτηρίζει τις παραπάνω καμπύλες είναι η Olog(N), όπου N ο αριθμός των στοιχείων που έχουν εισαχθεί στο δέντρο.

Σχόλια: Στον κώδικα υπολογίζεται και ο χρόνος εκτέλεσης των αναζητήσεων για κάθε μέθοδο ξεχωριστά. Τρέχοντας το πρόγραμμα 10 φορές τις 4 πιο γρήγορο ήταν το KD-Tree, ενώ το Quad-Tree βγήκε 6 φορές γρηγορότερο. Γενικά, και οι δύο δομές χαρακτηρίζονται από την έκφραση της πολυπλοκότητας Olog(N), η οποία αναφέρεται στον αναμενόμενο αριθμό βημάτων που απαιτούνται για την εκτέλεση μιας αναζήτησης σε ένα δέντρο με N κόμβους. Έτσι, είναι αναμενόμενο το PR Quad-Tree να είναι πιο γρήγορο από το KD-Tree, καθώς η πολυπλοκότητα για το μέγιστο βάθος στο οποίο φτάνουν υπολογίζεται 8.3 για το PR Quad-Tree και 16.6 για το KD-Tree.

Παρόλα αυτά, και οι δύο δομές δεδομένων χρησιμοποιούνται για την οργάνωση και την αναζήτηση δεδομένων και η απόδοση της κάθε μίας εξαρτάται από την φύση των δεδομένων και τον τρόπο χρήσης τους. Τα KD-Trees μπορούν να είναι πιο αποδοτικά για αναζήτηση σειράς δεδομένων, που έχουν ομοιόμορφη κατανομή και σε περιπτώσεις που ο χώρος διαχωρίζεται καλά από τους άξονες. Από την άλλη πλευρά, τα PR Quad-Trees αποδίδουν καλύτερα σε δεδομένα με ανισόμορφη κατανομή και για περιπτώσεις όπου ο χώρος δεν διαχωρίζεται καλά με γραμμικούς άξονες και για αυτό μπορούν να διαχειρίζονται αποτελεσματικότερα δυναμικές αλλαγές στα δεδομένα, στην εύρεση κοντινότερου γείτονα(find nearest neighbor) και εύρους(range). Στην περίπτωση της άσκησης η κατανομή των δεδομένων M = {200, 500, 1000, 10000,30000, 50000, 70000, 100000} είναι ανομοιόμορφη, γιατί η απόσταση μεταξύ των στοιχείων αυξάνεται όσο αυξάνεται η τιμή. Συνεπώς, αποδοτικότερη πρέπει να είναι η δομή δεδομένων PR Quad-Tree.