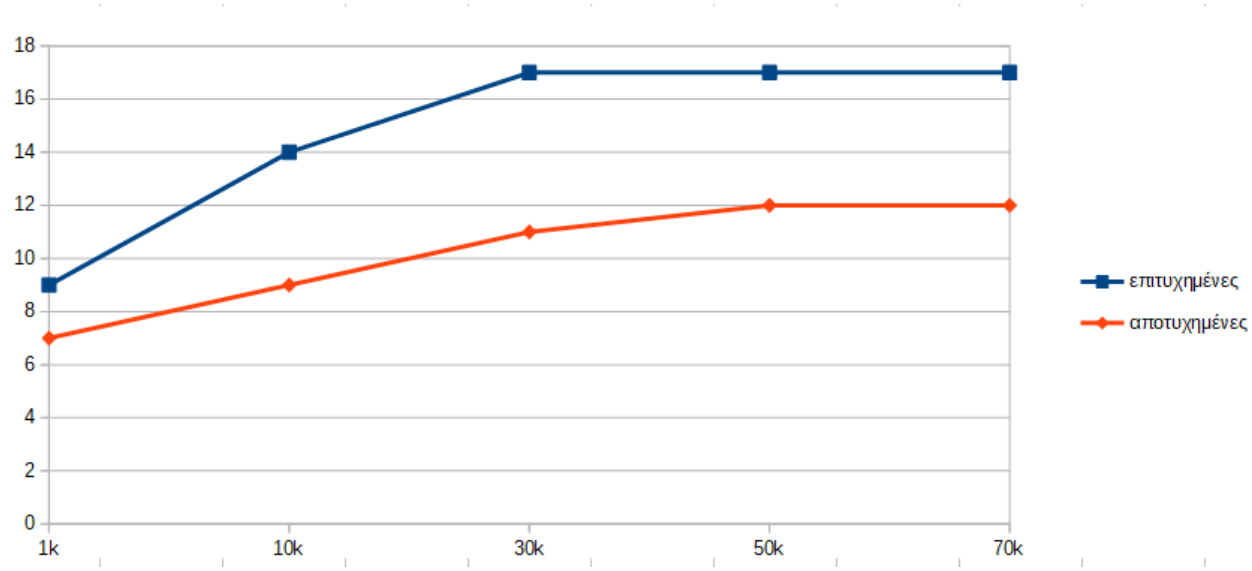
Εκθεση 2ης Ασκησης
Δομες Δεδομένων και αρχείων

Για το KD δεντρο:

Για να δημιουργήσω το Kd tree που ζητήθηκε αρχικά δημιουργήθηκε μια κλάση kdtree η οποία μέσα της έχει μία private κλάση TreeNode. Η TreeNode περιέχει τις τιμές του χ και y, και δύο TreeNode (left και right). Δημιουργώ στο tree ένα root και το αρχικοποιώ στο μηδέν ώστε να είναι κενό το δέντρο όταν δημιουργώ αντικείμενο για πρώτη φορά. Από μεθόδους έχω, την add και την search, οι οποίες χρησιμοποιούνται για να καλέσουν τις άλλες δύο private μεθόδους(λόγω των ορισμάτων τους) insertNode και searchNode. Η insertNode είναι μία αναδρομική συνάρτηση όπου παίρνει ορίσματα το array με τις τιμές , το root, και το depth(που ξεκινάει από μηδέν και κάθε φορά που καλεί τον εαυτό της αυξάνεται). Μέσω μιας βοηθητικής μεταβλητής (levels), βρίσκω το depth%2 που μου καθορίζει ώς προς ποια τιμή του array θα πρέπει να συγκρίνω και κατα συνέπεια, προς τα που θα συνεχιστεί η αναζήτηση. Η searchNode είναι επίσης αναδρομική , που μου επιτρέπει true/false ανάλογα με το αν βρήκε τον κόμβο. Αρχικά υπάρχει συνθήκη εξόδου ότι ο κόμβος είναι κενός (δεν μπορεί να πάει σε παρακάτω άρα δεν υπάρχει), και στην συνέχεια, πάλι με το levels και την ίδια διαδικασία με την insertNode, καλεί τον εαυτό της με το δεξί η το αριστερό παιδί του κόμβου που βρίσκεται. Τέλος γυρνάει true μόνο αν οι τιμές του κόμβου είναι ίδιες με τις δοσμένες. Στο κομμάτι της main, επειδή η πιθανότητα να βρεθεί τυχαία συνδυασμός χ,y που να υπάρχει μέσα στον κομβο είναι σχεδόν απίθανο, μέσω ενός πίνακα κρατάω τις τιμές των κόμβων ώστε να γίνουν (μέσω μίας for) οι επιτυχείς αναζητήσεις . Αντίθετα στις ανεπιτυχείς απλά μέσω μίας for απλά μπαίνουν μέσω της rand τυχαίες τιμές προς αναζήτηση.

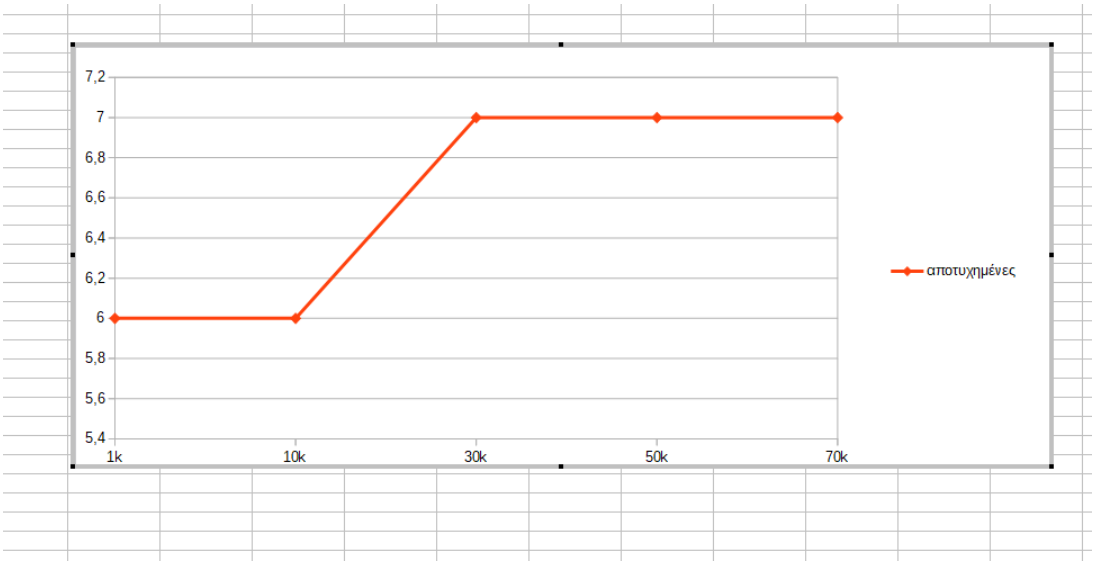
Διάγραμμα:



Παρατηρούμε ότι, η πολυπλοκότητα που παρουσιάζεται μέσω των αναζητήσεων (επιτυχημένων και αποτυχημένων), είναι log με βάση το δύο του n, όπου n είναι ο αριθμός των δεδομένων που μπαίνουν στο δέντρο.

Για το QuadTree:

Για την υλοποίηση του Quadtree έγιναν 3 κλάσεις. Μια Node, μια Point, και η Quad. Στην Point κρατουσα τα δεδομένα που ήθελα να προσθέσω η να αναζητήσω στο δέντρο. Στην Quad, με την δημιουργία ενός αντικειμένου δημιουργούνταν 4 “παιδιά” που αρχικοποιούνταν στο null, καθώς και οι διαστάσεις ώστε να πετύχω το NXN. Στην συνέχεια, μέσω μιας αναδρομικής insertNode πρόσθετα τον νέο κόμβο που είχε μέσα τον point με τα στοιχεία που θέλω να προσθέσω. Μετά έχω τη search, οι οποία, πάλι αναδρομικά, ψάχνει έναν Point για να διαπιστώσει αν βρίσκεται στο δέντρο η όχι. Για να γίνει αυτό, αρχικά έχω την συνθήκη εξόδου,ώστε να έχω επιστροφή null αν ο κόμβος δεν υπάρχει , ή να μας επιστρέφει τον κόμβο σε περίπτωση που τον βρει. Με αλλα λόγια, η τελευταία συνάρτηση που διαθέτει η quad μου (inboundary), απλώς ελέγχει αν το χ του κόμβου που θέλω να προσθέσω είναι ανάμεσα στα όρια της διάστασης NχN η όχι, και στην συνέχεια με το y κατ’ αντιστοιχία. Αν η παραπάνω συνθήκη επαληθεύεται, τότε επιστρέφει true αλλίως false. Για την επαλήθευση του δέντρου όπως ζητείται, στην main με δύο for-loop και τρόπο αντίστοιχο με το kdtree (δηλαδή με πίνακα αποθήκευσης τιμών για τις τιμές που μπαίνουν μέσα στο δέντρο και rand για τις υπόλοιπες) παίρνουμε τις επιθυμητές αναζητήσεις και στο τέλος βγάζουμε τον μέσο όρο τους. Σημαντικό να σημειώθει ότι για το επιτευγμα του NXN στην main() συνάρτηση έχουμε ορίσει στην δημιουργία του τον αριθμό των N που μας αναγράφεται.



Στις επιτυχείς αναζητήσεις του Quadtree παρουσιάστηκε πρόβλημα καθώς ενώ στην θεωρία έπρεπε κάθε φορά να λαμβάνω τον λογάριθμο των δεδομένων μου με βάση το τέσσερα παρουσιαζόταν σταθερός αριθμός.

Συμπεράσματα: Από τις αναζητήσεις των δύο διαφορετικών δέντρων kdtree, quadtree (επιτυχημένων – αποτυχημένων) παρατηρούμε ότι για ίδιο αριθμό δεδομένων το quadtree κάνει λιγότερες αναζητήσεις απο το kdtree. Αυτό αποδεικνύεται από τους λογαρίθμους που αντιστοιχούν στις παραπάνω μεθόδους ταξινόμησης δεδομένων(log, με βάση το 2 για το kdtree, με 4 για το quadtree). Συνεπώς για όλο και περισσότερα δεδομένα από τις δύο επιλογές θα προτιμηθεί το quadtree.

Πηγές: https://www.geeksforgeeks.org/quad-tree/
Απόλυτη Java, δημήτρης Ιακωβίδης.(ιδέα για την δημιουργία του Kdtree)