

Η εργασία είναι γραμμένη σε python. Χρειάζεται να εγκαταστήσετε τα πακέτα pandas και tqdm για να τρέξετε τον κώδικα.

Το 'dataset' που χρησιμοποιείται για τα tests είναι random αριθμοί, μεσα σ' ένα δωθεν range.

Τα δέντρα έχουν υλοποιηθεί σαν linked nodes, και όλη η λογική της λειτουργίας τους βρίσκεται μέσα στην κλαση-κομβο. Υπάρχει μια βοηθητική κλάση "Tree" που διασφαλίζει απλά ότι δε θα υπάρξει περιέργη συμπεριφορά σε άδεια δέντρα, περιπτώσεις που θα διαγραφεί η ριζά K.O.K. Πέρα απ' αυτό όλα τα υπόλοιπα calls περνάνε στη ριζά και προσπελάσουν recursively το δέντρο όπως χρειαστεί.

Insert

Απλή υλοποίηση, το στοιχείο που εισάγεται 'κατεβαίνει' το δέντρο, ανάλογα με τους κανόνες της κάθε μορφής (dimension ή quadrants), και μόλις βρεθεί κενό σημείο, εισάγεται.

Build

Σχετικά ακριβή πράξη, σε κάθε βήμα χωρίζει τη λίστα των σημείων που δόθηκαν, σε 2(kd) ή 4(quad) κομμάτια, τα οποία προωθεί στο αντίστοιχο κλαδί. Αν στο κλαδί δεν υπάρχει point, διαλέγεται από τη λίστα(κομμάτι) το σημείο πιο κοντά στο median, εισάγεται, και τότε προωθείται η λίστα σε αυτό.

Delete

Εδώ οι υλοποιήσεις διαφέρουν λίγο. Στα kd βρίσκουμε το point που βρίσκεται πιο κοντά στο σημείο διαγράψης, κατά τη διάσταση που χωρίζει, και το αντικαθιστούμε με αυτό. Συνεχίζεται επαναληπτικά μέχρι να βρεθεί φύλλο όπου απλά διαγράφεται.

Στα quad-trees δε βρήκα κάποια καλή υλοποίηση και αυτή που έχω κάνει δε μ' αρέσει. Βρίσκουμε το σημείο που πρέπει να διαγραφεί, αφαιρούμε όλο το sub-tree με ριζά αυτό το σημείο, και μετά επανεισάγουμε όλα τα στοιχεία.

Storage

trivial

Search

trivial

kNN – Search

Το ένα σημείο που αξίζει αναφορά εδώ είναι τα bounding boxes, που δημιουργούνται κατά την εκτέλεση. Όσο η συνάρτηση προσπελαύνει το δέντρο, κρατάει την περιοχή την οποία περικλείει (ή επιτρέπει/καθορίζει?) το κάθε κλαδί, on the fly. Μετά η υλοποίηση είναι σχετικά τυπική όπου αφότου βρεθούν k σημεία, συγκρίνουμε το "worst best-distance" με την απόσταση από την περιοχή που ορίζει το κάθε (επόμενο/άλλο) κλαδί, και μόνο αν αυτή είναι καλύτερη μπαίνουμε σ αυτό το για να ελέγξουμε για σημεία.