

Αναφορά Project Πολυδιάστατων Δομών Δεδομένων

Για την σύνταξη των δομών KD-Tree και Quad-Tree, έχουμε χρησιμοποιήσει την βιβλιοθήκη `recordtype`, προκειμένου να δημιουργήσουμε τους κόμβους των δέντρων. Το `recordtype` αποτελεί μια παραλλαγή της δομής `namedtuple`, που σε αντίθεση με την δομή `namedtuple`, έχει την δυνατότητα επεξεργασίας των πεδίων της κατά το runtime. Το KD-Tree είναι κόμβο-προσανατολισμένο, ενώ το Quad-Tree είναι φύλλο-προσανατολισμένο. Οι συναρτήσεις έχουν υλοποιηθεί σε μεμονωμένα αρχεία, τα οποία περιέχουν σχόλια. Τα αρχεία αυτά καλούνται σε ένα `main` αρχείο για κάθε δέντρο, προκειμένου να ελεγχθεί η λειτουργία των συναρτήσεων.

Μαζί με την αναφορά σας παραθέτουμε τα αρχεία `.py` με τις συναρτήσεις `build`, `insert`, `search`, `KNNsearch`, `rebalance`, `delete` για το KD-Tree και τις συναρτήσεις `build`, `insert`, `search`, `rebalance`, `delete`, `KNNsearch` για το Quad-Tree, καθώς και τα `main` αρχεία για τα δύο δέντρα.

Οι συναρτήσεις, καθώς και τα δέντρα έχουν φτιαχτεί ώστε να δέχονται σημεία της μορφής (X, Y) , όπου τα X, Y μπορεί να είναι αριθμητικές τιμές και αλφαριθμητικά. Οι συναρτήσεις `KNNsearch` και στα δύο δέντρα έχουν υλοποιηθεί ώστε να επιστρέφουν το πολύ K κοντινότερα σημεία, ελαχιστοποιώντας την Ευκλείδεια Απόσταση των K σημείων από το δεδομένο σημείο `QueryPoint`.

Παρακάτω παρουσιάζονται οι πολυπλοκότητες των αλγορίθμων που υλοποιήσαμε:

KD Tree	Υλοποίηση	Μέση Περίπτωση
<code>build</code>	$O(n^2 * \log_2(n))$	$O(n * \log_2(n))$
<code>insert</code>	$O(\log_2(n))$	$O(\log_2(n))$
<code>search</code>	$O(\log_2(n))$	$O(\log_2(n))$
<code>rebalance</code>	$O(n^3 * \log_2(n))$	$O(n^2 * \log_2(n))$
<code>delete</code>	$O(\log_2(n))$	$O(\log_2(n))$
<code>KNNsearch</code>	$O(n)$	$O(\log_2(n)^2 + k)$

Quad Tree	Υλοποίηση	Μέση Περίπτωση
<code>build</code>	$O(n^2 * \log_4(n))$	$O(n * \log_4(n))$
<code>insert</code>	$O(\log_4(n))$	$O(\log_4(n))$
<code>search</code>	$O(\log_4(n))$	$O(\log_4(n))$
<code>rebalance</code>	$O(n^3 * \log_4(n))$	$O(n^2 * \log_4(n))$
<code>delete</code>	$O(\log_4(n))$	$O(\log_4(n))$
<code>KNNsearch</code>	$O(n)$	$O(\log_4(n)^2 + k)$

Χρόνοι εκτέλεσης για το KD tree σε seconds:

Kd Tree Build: 0.008975982666015625

Kd Tree Search: 0.016955852508544922

Kd Tree Delete: 0.0019953250885009766

Kd Tree KNNSearch: 0.4888608455657959

Χρόνοι εκτέλεσης για το Quad tree σε seconds:

Quad Tree Build: 0.021941423416137695

Quad Tree Search: 0.016954660415649414

Quad Tree Delete: 0.018949031829833984

Quad Tree KNNSearch: 3.9610228538513184

Μέλη ομάδας:

Όμηρος Βασδάρης AM: 1054429

Δημήτριος Κωστορρίζος AM:1054419

Νικόλαος Μπακάλης AM: 1054316

Ζήσης Στυλιανός Τράμπαρης AM: 1054355