



# Project Πολυδιάστατες Δομές Δεδομένων και Υπολογιστική Γεωμετρία

ΠΕΙΡΑΜΑΤΙΚΗ ΣΥΓΚΡΙΣΗ KD-TREES ΚΑΙ QUAD-TREES

Απόστολος Ντεληδήμος | 1047199 | 5<sup>ο</sup> Έτος  
Δημήτριος Τσότσουλας | 1047197 | 5<sup>ο</sup> Έτος

Η γλώσσα που υλοποιήθηκε το project ήταν η Python. Το IDE που χρησιμοποιήσαμε ήταν το Visual Studio Code. Για την πειραματική σύγκριση των δομών χρησιμοποιήθηκε ένα artificial dataset των 5000 2d δεδομένων.

Δοκιμάσαμε και synthetic dataset των 100.000 δεδομένων για καλύτερα αποτελέσματα ωστόσο υπήρξαν προβλήματα με το IDE και επιλέξαμε μικρότερο αριθμό δεδομένων.

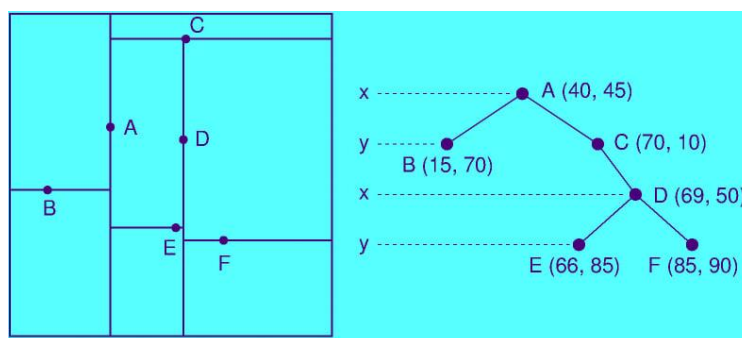
Για τις μετρήσεις μας ,χρησιμοποιήθηκε η βιβλιοθήκη timeit της Python. Οι εντολές ήταν οι : start = timer() και end = timer() , print(end-start).

Οι υλοποιήσεις των δέντρων βρίσκονται στα αρχεία KDTree.py και QuadTree.py.

Οι υλοποίηση των δοκιμών έγινε στα αρχεία testkdtree.py και testquadtree.py.

## KD-TREES

Τα KD-Trees είναι δυαδικά δέντρα που δημιουργούνται διαμερίζοντας αναδρομικά το σύνολο δεδομένων ως προς κάποια διάσταση. Για την εισαγωγή ενός στοιχείου ξεκινάμε από τη ρίζα και ανάλογα με το επίπεδο που βρισκόμαστε ,συγκρίνουμε ως προς κάποια διάσταση x ή y και ακολουθούμε τους αντίστοιχους δείκτες.



Παρακάτω θα αναλυθεί η απόδοση των KD-Trees στις πράξεις Build, Insert, Delete, Update και Search.

## -BUILD

Αρχικά δημιουργούμε με την εντολή create ένα κενό δέντρο 2-διαστάσεων σε χρόνο: 7.35110000000816e-05 sec

```
tree = KD.create(dimensions=2)
```

## -INSERT

Με την εντολή insert εισάγουμε στο δέντρο 5010 2d δεδομένα σε χρόνο: 0.36 sec δημιουργώντας το kd-tree.

```
for x in range(5010):  
    tree.insert([int(results[x][0]),int(results[x][1])])
```

Ο χρόνος εισαγωγής ενός στοιχείου είναι 6.629799999924302e-05 sec

## -DELETE

Πρέπει να ελέγχουμε αν το δέντρο είναι ισορροπημένο και να το κάνουμε rebalance άμα χρειαστεί.

```
tree.delete([int(results[3739][0]),int(results[3739][1])])  
if tree.is_balanced == False:  
    tree = tree.rebalance()  
print(tree)
```

Η διαγραφή ενός στοιχείου γίνεται σε χρόνο 0.0006835069999997501 sec

Η διαγραφή πολλών στοιχείων γίνεται σε χρόνο: 1.5918307059999997 sec

## -UPDATE

## -SEARCH

- KNN

Με την εντολή search\_knn ,δίνουμε ένα point στην συνάρτηση και μας εμφανίζει τους k nearest neighbors. Στην πρώτη περίπτωση έχουμε τον χρόνο για το στοιχείο 844535 ,424642 για k=10 :

```
print(tree.search_knn([844535, 424642] ,10))
```

Εμφανίζοντας τους 10 γείτονες του σε χρόνο: 0.002 sec

Εκτελώντας την ίδια εντολή για k=3346 γείτονες έχουμε χρόνο: 0.10 sec

```
print(tree.search_knn([844535, 424642] ,3346))
```

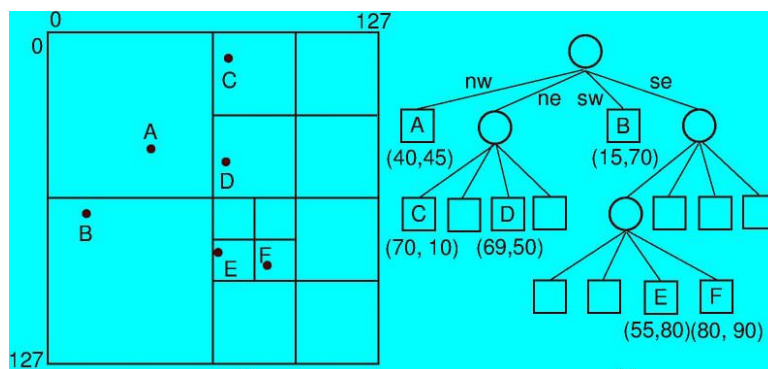
Επιπλέον με την εντολή `search_nn` μπορούμε να ελέγξουμε αν ένα στοιχείο υπάρχει στο δέντρο. Αν η απόσταση που πάρουμε είναι 0, σημαίνει ότι το στοιχείο υπάρχει στο δέντρο. Πιο απλά, κάνουμε αναζήτηση KNN για k=1

```
print(tree.search_nn([322365,530808]))
```

```
(<KNode - [322365, 530808]>, 0.0)
```

## QUAD-TREES

Τα Quad-Trees χρησιμοποιούνται για δεδομένα 2-διαστάσεων. Στηρίζονται στις αρχές της περιοδικά επαναλαμβανόμενης διάσπασης. Κάθε κόμβος-γονέας έχει 4 απογόνους και ο χώρος διασπάται σε 4 ίσου μεγέθους τεταρτημόρια.



Παρακάτω θα αναλυθεί η απόδοση των Quad-Trees στις πράξεις Build, Insert, Delete, Update και Search.

### -BUILD

Αρχικά δημιουργούμε ένα κενό δέντρο 2-διαστάσεων σε χρόνο:  
2.7462000000033626e-05 sec

```
qt = Node(0, 0, 100000000000, 100000000000)
```

### -INSERT

Με την εντολή insert εισάγουμε στο δέντρο 5010 2d δεδομένα σε χρόνο:  
0.1958258369999999 sec δημιουργώντας το quad-tree.

```
for x in range(5000):  
    qt.insert(rects[x])
```

Ο χρόνος εισαγωγής ενός στοιχείου είναι 3.744899999991169e-05 sec.

### -DELETE

Η διαγραφή πολλών 5010 στοιχείων γίνεται σε χρόνο:  
0.023671973999999985 sec

Η διαγραφή ενός στοιχείου γίνεται σε χρόνο: 0.00012677099999991448 sec

```
for x in range(5010):  
    qt.delete(rects[x])
```

### -UPDATE

Το update ενός στοιχείου γίνεται σε χρόνο: 0.00013065399999998968 sec

## **-SEARCH**

### **-KNN**

Ο κώδικας που προσπαθήσαμε να υλοποιήσουμε ήταν εσφαλμένος και δεν το συμπεριλάβαμε

## **ΣΥΓΚΡΙΣΗ ΤΩΝ KD-TREES ΚΑΙ QUAD-TREES**

	<b>KD-TREES</b>	<b>QUAD-TREES</b>
<b>BUILD</b>	7.351100000000816e-05	2.74620000000033626e-05
<b>INSERT(1)</b>	6.629799999924302e-05	3.744899999991169e-05
<b>INSERT(5010)</b>	0.36	0.1958258369999999
<b>DELETE(1)</b>	0.0006835069999997501	0.00012677099999991448
<b>DELETE(5010)</b>	1.5918307059999997	0.023671973999999985
<b>UPDATE</b>	-	0.0001306539999998968
<b>KNN-SEARCH(10)</b>	0.002	-