

**Projects: Μελέτη-Υλοποίηση και Πειραματική Αξιολόγηση
Πολυδιάστατων Δομών Δεδομένων με Εφαρμογές τους-**

Project-1:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΑΡΙΔΑΚΗΣ ΓΕΩΡΓΙΟΣ **ΑΜ:**1072478 **ΕΤΟΣ:**5ο

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΚΑΓΚΟΣ ΙΩΑΝΝΗΣ **ΑΜ:**1072611 **ΕΤΟΣ:**5ο

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΣΤΕΡΓΙΟΥ ΓΕΩΡΓΙΟΣ **ΑΜ:**1072503 **ΕΤΟΣ:**5ο

ΠΑΡΑΔΟΧΗ: Τα screenshot για τα μεγάλα δέντρα είναι ενδεικτικά (το πρώτο και το τελευταίο) λόγω του μεγάλου μεγέθους των αποτελεσμάτων

ΠΕΡΙΓΡΑΦΗ:

Η υλοποίηση της εργασίας πραγματοποιήθηκε σε γλώσσα προγραμματισμού Python στο περιβάλλον IntelliJ. Χρησιμοποιήθηκαν τα modules: pandas για ανάγνωση csv αρχείου,math για μαθηματικές εξισώσεις, timeit για τη χρονομέτρηση διαδικασιών, datasketch για την hash υλοποίηση.

Υλοποιήσαμε κώδικα για την αυτόματη δημιουργία ενός αρχείου csv από το σύνδεσμο της Wikipedia:https://en.wikipedia.org/wiki/List_of_computer_scientists με τα columns Surname,#Awards,Education,#DBLP_Record το ExportWikiData.py

Καθώς δεν καταφέραμε να εντοπίσουμε κάποιο μοτίβο ώστε να αφαιρέσουμε τα υπολοιπά ονόματα και να κρατήσουμε μόνο το surname με αυτόματο τρόπο τα αφαιρέσαμε με manually τρόπο στο αρχείο csv που εξάγει το ExportWikiData.py.

	A	B	C	D	E	F	G	H	I
1	Surname	#Awards	Education	#DBLP_Record					
2	Atta	10	Khan was a Bright Sparks scholar	71					
3	Aalst	0		1,119					
4	Aaronson	7		230					
5	Abebe	9		41					
6	Abelson	5	Abelson graduated with a	53					
7	Abiteboul	2	The son of two hardware store ow	317					
8	Abramsky	6	Abramsky was educated at Hasm	215					
9	Adleman	2		54					
10	Agrawal	9		105					
11	Ahn	2		40					
12	Alfred	0		113					
13	Allen	5		37					
14	Amdahl	4		16					
15	Anderson	0	Anderson received a BA in mather	573					
16	Anthony	6	Lisa Anthony earned her B.S. and	148					
17	Appel	0		148					
18	Aragon	0	Aragon received her B.S. In mathe	83					
19	Arden	0		37					
20	Angie	0		1					
21	Arora	0		245					
22	Asprey"	0		0					
23	Atanasoff	0		46					
24	Atre	0		0					
25	Babbage	0		34					
26	Bachman	5		77					
27	Backous	0		71					
28	Backus	9		25					
29	Bacon	0		124					
30	Bader	0		310					
31	Bahl	17		129					
32	Barr	0		18					
33	Bartik	4		0					
34	Barto	2		219					
35	Bauer	14		196					
36	Bayer	0		89					
37	Bell	0		162					

A	B	C	D	E	F	G	H	I
650 Westervelt	0		2					
651 Whittaker	5		208					
652 Widom	0	Widom earned a BS degree in trun	219					
653 Wiederhol	0		207					
654 Wiener	9		34					
655 Wijngaard	0		49					
656 Wilkes	0		2					
657 Wilkes	0		84					
658 Wilks	19		211					
659 Wilkinson	5	Born in Strood, England, he won a	47					
660 Wilson	1		18					
661 Winograd	0		40					
662 Winograd	5	Winograd grew up in Colorado ar	122					
663 Winston	0		61					
664 Wirth	0		115					
665 Wiseman	0		29					
666 Wisnosky	0		0					
667 Wolfram	0	Wolfram was educated at Eton Co	56					
668 Woodger	0		0					
669 Woodward	0		26					
670 Worsley	5		1					
671 Wozniak	14		13					
672 Wu	0		7,065					
673 Wulf	0		150					
674 Yannakakis	1		291					
675 Yao	1		179					
676 Yen	0		419					
677 Yoneda	0		1					
678 Yourdon	0		22					
679 Yung	12		748					
680 Zadeh	17		237					
681 Zantema	0		118					
682 Zaman	0		35					
683 Zdonik	0	Zdonik has received a number of r	194					
684								
685								
686								

BigDataset



Καθώς και ένα μικρότερο αρχείο csv για τη διευκόλυνση υλοποίησης του project
Dataset.csv

A1	Surname	#Awards	Education	#DBLP_Record	
1	Atta	10	Khan was	71	
2	Aalst	0		1,119	
3	Aaronson	7		230	
5	Abebe	9		41	
6	Abelson	5	Abelson	53	
7	Abiteboul	2	The son of	317	
8	Abramsky	6	Abramsky	215	
9	Adleman	2		54	
10	Agrawal	9		105	
11	Ahn	2		40	
12					

Η μέθοδος LSH:

Η Lsh method δημιουργήθηκε από κοινού και χρησιμοποιήθηκε η ίδια για τα δέντρα Range-Tree, KD-Tree με τις εξής συναρτήσεις:

create_minhash: Το text είναι ένα string που αντιπροσωπεύει το κείμενο για το οποίο θέλουμε να δημιουργήσουμε ένα minhash. Η συνάρτηση αρχικοποιεί ένα αντικείμενο Minhash (Minhash) με 128 μεταβολές. Για κάθε λέξη, ενημερώνει το αντικείμενο minhash κωδικοποιώντας τη λέξη ως UTF-8 και ενημερώνοντας το minhash. Η συνάρτηση επιστρέφει το δημιουργημένο αντικείμενο minhash.

Lsh_education_similarity: Αρχικοποιεί ένα όριο που βασίζεται στο δεδομένο ποσοστό ομοιότητας. Στη συνέχεια δημιουργεί αντικείμενα minhash για τον τομέα της «εκπαίδευσης» κάθε εγγραφής χρησιμοποιώντας τη λειτουργία *create_minhash*. Αυτά τα αντικείμενα minhash χρησιμοποιούνται για τη δημιουργία ενός δείκτη LSH (LSH). Κάθε αντικείμενο Minhash εισάγεται στον δείκτη LSH.

Ο δείκτης LSH ερωτάται για παρόμοια στοιχεία με βάση τα minhashes. Εάν δύο minhashes έχουν μια ομοιότητα jaccard μεγαλύτερη από το όριο, θεωρούνται παρόμοια. Τέλος, η συνάρτηση επιστρέφει ως έξοδο μια λίστα εγγραφών που αναγνωρίζονται ως παρόμοια με βάση το δείκτη LSH.

KD-TREE: Η παρακάτω υλοποίηση αφορά την κατασκευή και επεξεργασία ενός KD-tree 3 διαστάσεων το οποίο περιέχει τα δεδομένα ενός πλήθους επιστημόνων, δηλαδή το όνομα, τον αριθμό των βραβείων και των αριθμό του DBLP ιστορικού τους στις x,y και z διαστάσεις αντίστοιχα, τα οποία αντλούνται από ένα csv αρχείο. Κάθε κόμβος αποθηκεύει τις πληροφορίες για κάθε επιστήμονα που απαιτούνται στις αντίστοιχές διαστάσεις και τα δεδομένα αυτά αξιοποιούνται για την αναζήτηση των κόμβων. Υλοποιούνται οι βασικές πράξεις Δημιουργίας, Εισαγωγής, Διαγραφής, Ενημέρωσης και Αναζήτησης κόμβων που αναλύονται παρακάτω. Ο χρήστης καλείται να επιλέξει ανάμεσα σε 7 λειτουργίες όσον αφορά την επεξεργασία του δέντρου και να δώσει τις αντίστοιχες ζητούμενες εξόδους.

ΛΕΙΤΟΥΡΓΙΕΣ:

Δημιουργία δέντρου: Η μέθοδος make_kd_tree είναι μια συνάρτηση που κατασκευάζει ένα k-διάστατο δέντρο (KD-tree) από μια λίστα σημείων σε ένα χώρο k διαστάσεων. Ένα KD-tree είναι μια δομή δεδομένων που διαμερίζει τον χώρο σε περιοχές για τη βελτιστοποίηση λειτουργιών αναζήτησης. Στην περίπτωση μας το KD-tree χτίζεται σε έναν χώρο 3 διαστάσεων (x, y, z). Πιο συγκεκριμένα τα ορίσματα εισόδου της είναι μια λίστα στοιχείων x,y,z (που αντιπροσωπεύουν τα Ονόματα, πλήθος Βραβείων και πλήθος ιστορικού DBLP), ο αριθμός των k-διαστάσεων και το βάθος για κάθε κόμβο.

Εάν η είσοδος λίστας σημείων είναι κενή, η μέθοδος επιστρέφει None, παριστάνοντας ένα κενό δέντρο. Εάν η λίστα δεν είναι κενή δημιουργείται ένας νέος κόμβος (κλάση Node) με τις συντεταγμένες του μεσαίου σημείου. Τα εύρη x, y και z για τον κεντρικό κόμβο ορίζονται με βάση τις ελάχιστες και μέγιστες τιμές σε κάθε διάσταση. Η μέθοδος καλείται αναδρομικά στα αριστερά και δεξιά μισά των ταξινομημένων σημείων. Η αναδρομή κλίση της συνεχίζεται μέχρι να φτάσει στη βάση, με αποτέλεσμα τη δημιουργία του KD-tree. Κάθε κόμβος στο δέντρο αντιπροσωπεύει ένα σημείο στον χώρο, και η δομή του δέντρου έχει ως αποτέλεσμα γρηγορότερες αναζητήσεις για τα στοιχεία της αρχικής λίστας. Τέλος η make_kd_tree καλεί την printTree (περισσότερες λεπτομέρειες παρακάτω) ώστε να δούμε το παραγόμενο δέντρο.

Εκτύπωση δέντρου: Η συνάρτηση printTree, η οποία δέχεται σαν όρισμα τη ρίζα του δέντρου και ένα κενό, τυπώνει το παραγόμενο δέντρο. Πιο συγκεκριμένα, διατρέχει το δέντρο με βάση το βάθος και εκτυπώνοντας τις πληροφορίες κάθε κόμβου που συναντά. Οι πληροφορίες που εκτυπώνονται περιλαμβάνουν τις συντεταγμένες (x, y, z) του κόμβου και τα εύρη x, y, και z που σχετίζονται με αυτόν τον κόμβο. Στη συνέχεια καλεί αναδρομικά τον εαυτό της για το αριστερό και το δεξιό παιδί - κόμβο του τρέχοντος κόμβου. Αυτή η αναδρομή συνεχίζεται μέχρι να φτάσει στα φύλλα του δέντρου δηλαδή όταν δεν υπάρχει άλλος κόμβος.

Στιγμιότυπο για το μικρό δέντρο:

```
Choose one of the above by its number:  
X: Abramsky Y:6 Z:215.0 Range X: Aalst - Atta Range Y: 0 - 10 Range Z: 71.0 - 317.0  
-leftchild-X:Abelson Y:5 Z:55.0 Range X: Aalst - Abiteboul Range Y: 0 - 9 Range Z: nan - 317.0  
-leftchild--leftchild-X:Abiteboul Y:2 Z:317.0 Range X: Aalst - Abiteboul Range Y: 0 - 2 Range Z: nan - 317.0  
-leftchild--leftchild-leftchild-X:Aalst Y:0 Z:nan Range X: Aalst - Aalst Range Y: 0 - 0 Range Z: nan - nan  
-leftchild--rightchild-X:Aaronson Y:7 Z:230.0 Range X: Aaronson - Abebe Range Y: 7 - 9 Range Z: 41.0 - 230.0  
-leftchild--rightchild--leftchild-X:Abebe Y:9 Z:41.0 Range X: Abebe - Abebe Range Y: 9 - 9 Range Z: 41.0 - 41.0  
-rightchild-X:Agrawal Y:9 Z:105.0 Range X: Adleman - Atta Range Y: 2 - 10 Range Z: 40.0 - 105.0  
-rightchild--leftchild-X:Adleman Y:2 Z:54.0 Range X: Adleman - Ahn Range Y: 2 - 2 Range Z: 40.0 - 54.0  
-rightchild--leftchild-leftchild-X:Ahn Y:2 Z:40.0 Range X: Ahn - Ahn Range Y: 2 - 2 Range Z: 40.0 - 40.0  
-rightchild--rightchild-X:Atta Y:10 Z:71.0 Range X: Atta - Atta Range Y: 10 - 10 Range Z: 71.0 - 71.0  
  
Make tree executed in: 9.28000000000399e-05 seconds  
Give a number for each choice below:
```

Στιγμιότυπο για το μεγάλο δέντρο:

```

Choose one of the above by its number:
X-Ladner Y:0 Z:250.0 Range X\_Aalist\_Zonnik\_Range\_Y: 0 - 43 Range Z: 0.0 - 993.0
-leftchild-leftchild-X-Henry Y:0 Z:319.0 Range X: Aalist - Kötting Range Y: 0 - 38 Range Z: nan - nan
-leftchild-leftchild-X-Cohen Y:0 Z:280.0 Range X\_Aalist\_Helmer\_Range\_Y: 0 - 0 Range Z: nan - nan
-leftchild-leftchild-leftchild-X-Coldwater Y:0 Z:0.0 Range X\_Aalist\_Fulkerson\_Range\_Y: 0 - 0 Range Z: nan - 83.0
-leftchild-leftchild-leftchild-leftchild-X-Betvinnik Y:0 Z:0.0 Range X\_Aalist\_Codd\_Range\_Y: 0 - 0 Range Z: nan - 83.0
-leftchild-leftchild-leftchild-leftchild-leftchild-X-Barry Y:0 Z:18.0 Range X: Aalist - Booth Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild-leftchild-leftchild-leftchild-leftchild-X-Bezzy Y:0 Z:1.0 Range X\_Aalist\_Booth\_Range\_Y: 0 - 0 Range Z: nan - 14.0
-leftchild-leftchild-leftchild-leftchild-leftchild-X-Asprey Y:0 Z:0.0 Range X: Aalist - Berdichevsky Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-X-Aalist Y:0 Z:nan Range X: Aalist - Aalist Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Berdichevsky Y:0 Z:1.0 Range X: Aree - Berdichevsky Range Y: 0 - 0 Range Z: 0.0 - 1.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Atre Y:0 Z:0.0 Range X: Atre - Atre Range Y: 0 - 0 Range Z: 0.0 - 0.0
-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Bowick Y:0 Z:7.0 Range X: Blaave - Booth Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Blaave Y:0 Z:14.0 Range X: Blaave - Elm Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Blum Y:0 Z:1.0 Range X: Blum - Elm Range Y: 0 - 0 Range Z: 1.0 - 1.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Blum Y:0 Z:33.0 Range X: Booth - Booth Range Y: 0 - 0 Range Z: 3.0 - 3.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Bernard Y:0 Z:73.0 Range X: Blum - Booth Range Y: 0 - 0 Range Z: 23.0 - 83.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Aragon Y:0 Z:46.0 Range X: Aragon - Backhouse Range Y: 0 - 0 Range Z: 34.0 - 83.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Aragon Y:0 Z:33.0 Range X: Aragon - Aragon Range Y: 0 - 0 Range Z: 37.0 - 83.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Arden Y:0 Z:37.0 Range X: Arden - Arden Range Y: 0 - 0 Range Z: 37.0 - 37.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Backhouse Y:0 Z:71.0 Range X: Baggage - Backhouse Range Y: 0 - 0 Range Z: 34.0 - 71.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Babbage Y:0 Z:34.0 Range X: Babbage - Baggage Range Y: 0 - 0 Range Z: 34.0 - 34.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Ballacker Y:0 Z:24.0 Range X: Blushan - Booth Range Y: 0 - 0 Range Z: 23.0 - 50.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Bhushan Y:0 Z:32.0 Range X: Blushan - Blum Range Y: 0 - 0 Range Z: 23.0 - 32.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Blum Y:0 Z:23.0 Range X: Blum - Blum Range Y: 0 - 0 Range Z: 23.0 - 23.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Booth Y:0 Z:50.0 Range X: Booth - Booth Range Y: 0 - 0 Range Z: 50.0 - 50.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Church Y:0 Z:21.0 Range X: Bourne - Cambell Range Y: 0 - 0 Range Z: 0.0 - 21.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Campbell Y:0 Z:0.0 Range X: Bourne - Cambell Range Y: 0 - 0 Range Z: 1.0 - 21.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Brenesham Y:0 Z:1.0 Range X: Bourne - Cambell Range Y: 0 - 0 Range Z: 11.0 - 15.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Breed Y:0 Z:11.0 Range X: Breed - Breed Range Y: 0 - 0 Range Z: 11.0 - 11.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Brin Y:0 Z:21.0 Range X: Brin - Candin Range Y: 0 - 0 Range Z: 4.0 - 21.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Candin Y:0 Z:4.0 Range X: Candin - Candin Range Y: 0 - 0 Range Z: 4.0 - 4.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Chau Y:0 Z:0.0 Range X: Cantrell - Clermont Range Y: 0 - 0 Range Z: 0.0 - 13.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Cantrell Y:0 Z:13.0 Range X: Cantrell - Clermont Range Y: 0 - 0 Range Z: 5.0 - 13.0
-leftchild-leftchild-leftchild-leftchild-leftchild-leftchild-rightchild-leftchild-X-Carmack Y:0 Z:5.0 Range X: Carmack - Carmack Range Y: 0 - 0 Range Z: 5.0 - 5.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-X-Wingfield Y:5 Z:5 - 5 Range X: Warwick - Warwick Range Y: 5 - 5 Range Z: 1.0 - 124.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Warwick Y:5 Z:122.0 Range X: Warwick - Warwick Range Y: 5 - 5 Range Z: 47.0 - 124.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Watkinson Y:5 Z:47.0 Range X: Wilkinson - Wilkinson Range Y: 5 - 5 Range Z: 47.0 - 47.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Worsey Y:5 Z:1.0 Range X: Worsey - Worsey Range Y: 5 - 5 Range Z: 1.0 - 1.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Thalmann Y:5 Z:734.0 Range X: Srinari - Whiteman Range Y: 5 - 5 Range Z: 152.0 - 777.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Tamassia Y:4 Z:307.0 Range X: Srinari - Whiteman Range Y: 3 - 4 Range Z: 152.0 - 593.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Srinari Y:4 Z:152.0 Range X: Srinari - Whiteman Range Y: 4 - 4 Range Z: 282.0 - 593.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Tardos Y:3 Z:154.0 Range X: Srinari - Whiteman Range Y: 4 - 4 Range Z: 152.0 - 152.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Wegner Y:3 Z:154.0 Range X: Vardi - Whittaker Range Y: 3 - 5 Range Z: 152.0 - 777.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Vardi Y:3 Z:777.0 Range X: Vardi - Whittaker Range Y: 3 - 3 Range Z: 169.0 - 777.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Watler Y:3 Z:109.0 Range X: Watler - Watler Range Y: 3 - 3 Range Z: 169.0 - 169.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Watler Y:3 Z:109.0 Range X: Watler - Watler Range Y: 3 - 3 Range Z: 169.0 - 169.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Zoussaint Y:11 Z:213.0 Range X: Zoussaint - Zoussaint Range Y: 3 - 26 Range Z: 0.0 - 748.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Yapkin Y:8 Z:0.0 Range X: Shannon - Wozniak Range Y: 6 - 26 Range Z: 0.0 - 211.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Sutherland Y:12 Z:52.0 Range X: Shannon - Taylor Range Y: 7 - 26 Range Z: 15.0 - 219.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Taylor Y:7 Z:119.0 Range X: Tate - Taylor Range Y: 7 - 7 Range Z: 65.0 - 119.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Shannon Y:15 Z:43.0 Range X: Shannon - Stalman Range Y: 15 - 26 Range Z: 15.0 - 43.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-rightchild-X-Shannon Y:15 Z:11.0 Range X: Stalman - Stalman Range Y: 26 - 26 Range Z: 15.0 - 15.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-rightchild-X-Wozniak Y:14 Z:11.0 Range X: Vaughan - Wozniak Range Y: 7 - 19 Range Z: 0.0 - 211.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Vaughan Y:7 Z:10.0 Range X: Vaughan - Wozniak Range Y: 7 - 9 Range Z: 0.0 - 34.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Vaughan Y:7 Z:0.0 Range X: Vaughan - Vaughan Range Y: 7 - 7 Range Z: 0.0 - 0.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-rightchild-X-Wilks Y:19 Z:211.0 Range X: Wilks - Wilks Range Y: 19 - 19 Range Z: 211.0 - 211.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Tanenbaum Y:22 Z:219.0 Range X: Shabolt - Zadeh Range Y: 8 - 29 Range Z: 219.0 - 748.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-X-Shamir Y:11 Z:294.0 Range X: Shabolt - Sifakis Range Y: 10 - 15 Range Z: 242.0 - 514.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Shamir Y:11 Z:338.0 Range X: Shabolt - Sifakis Range Y: 10 - 11 Range Z: 242.0 - 338.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Sifakis Y:10 Z:242.0 Range X: Sifakis - Sifakis Range Y: 18 - 18 Range Z: 242.0 - 242.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Sifakis Y:10 Z:514.0 Range X: Sifakis - Sifakis Range Y: 15 - 15 Range Z: 514.0 - 514.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Yung Y:12 Z:748.0 Range X: Tarjan - Tarjan Range Y: 8 - 17 Range Z: 237.0 - 748.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Veloso Y:9 Z:627.0 Range X: Tarjan - Veloso Range Y: 8 - 9 Range Z: 350.0 - 627.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Tarjan Y:8 Z:350.0 Range X: Tarjan - Tarjan Range Y: 8 - 8 Range Z: 350.0 - 350.0
-rightchild-rightchild-rightchild-rightchild-rightchild-leftchild-leftchild-leftchild-leftchild-X-Zadeh Y:17 Z:237.0 Range X: Zadeh - Zadeh Range Y: 17 - 17 Range Z: 237.0 - 237.0

```

Make tree executed in: 0.009561500000000223 seconds

Αναζήτηση κόμβου: Η αναζήτηση ενός κόμβου γίνεται μέσω της searchKdTree() που δέχεται ως ορίσματα την ρίζα του δέντρου, τον κόμβο προς αναζήτηση, τη διάσταση του δέντρου και το βάθος του κόμβου. Η συνάρτηση χρησιμοποιεί τον άξονα για να καθορίσει ποια διάσταση θα ληφθεί υπόψη στο τρέχον βάθος. Στη συνέχεια, συγκρίνει τις συντεταγμένες του κόμβου-στόχου με τις συντεταγμένες του τρέχοντος κόμβου σε αυτήν τη διάσταση. Αν η συντεταγμένη του κόμβου-στόχου είναι μικρότερη από τη συντεταγμένη του τρέχοντος κόμβου στη τρέχον διάσταση, πηγαίνει στο αριστερό υποδέντρο. Αν η συντεταγμένη του κόμβου-στόχου είναι μεγαλύτερη ή ίση με τη συντεταγμένη του τρέχοντος κόμβου στη τρέχον διάσταση, πηγαίνει στο δεξί υποδέντρο. Η συνάρτηση συνεχίζει αυτήν τη διαδικασία μέχρι να βρει τον κόμβο-στόχο, τότε εκτυπώνει ένα μήνυμα που υποδεικνύει ότι ο κόμβος βρέθηκε εάν δεν τον βρει εκτυπώνει μήνυμα που ενημερώνει τον χρήστη ότι ο κόμβος δεν βρέθηκε.

Στιγμιότυπο για το μικρό δέντρο:

```
0) End the program
1) Create the K-Dimensional tree
2) Print the K-Dimensional tree
3) Insert a node
4) Search for a node
5) Delete a node
6) Update a node
7) Query search
Choose one of the above by its number:4
Insert the x value of node to be searched:Abebe
Insert the y value of node to be searched:9
Insert the z value of node to be searched:41.0
Search complete. Node found.
```

Node search executed in: 3.42000000017606e-05 seconds

Στιγμιότυπο για το μεγάλο δέντρο:

```
Give a number for each choice below:
0) End the program
1) Create the K-Dimensional tree
2) Print the K-Dimensional tree
3) Insert a node
4) Search for a node
5) Delete a node
6) Update a node
7) Query search
Choose one of the above by its number:4
Insert the x value of node to be searched:Tate
Insert the y value of node to be searched:7
Insert the z value of node to be searched:63
Could not find the node. Please try again.
```

Node search executed in: 4.179999996267725e-05 seconds

Εισαγωγή κόμβου: Η εισαγωγή ενός νέου κόμβου εκτελείται από τη συνάρτηση insertKdTree() που δέχεται ως ορίσματα τη ρίζα του δέντρου, τον νέο κόμβο, τη διάσταση του δέντρου και το βάθος της ρίζας. Αρχικά η συνάρτηση υπολογίζει τον άξονα που βρίσκεται ο κόμβος που κάλεσε την συνάρτηση και ελέγχει αν ο κόμβος αυτός έχει και αριστερό και δεξιό παιδί. Στην περίπτωση που ο κόμβος δεν έχει ένα από τα δύο παιδιά τότε ελέγχεται το εάν ο νέος κόμβος πληροί τις προϋποθέσεις για να γίνει παιδί του εκάστοτε υπάρχον κόμβου. Αν δεν εισαχθεί ο κόμβος η διαδικασία συνεχίζει καλώντας αναδρομικά την συνάρτηση είτε από το δεξιό είτε από το

αριστερό παιδί του κάθε κόμβου ώσπου ο κόμβος να εισαχθεί εντός του δέντρου και τερματίζει καλώντας την update_range() για ενημέρωση του εύρους κάθε κόμβου. Η update_range() δέχεται ως ορίσματα τη ρίζα του δέντρου, τη λίστα με όλα τα στοιχεία του δέντρου, τη διάσταση του δέντρου και το βάθος. Ακολουθεί παρόμοια διαδικασία με την δημιουργία δέντρου αλλά αντί να δημιουργεί κόμβους ενημερώνει τους υπάρχον. Τέλος καλείται αναδρομικά να ενημερώσει όλους τους κόμβους του δέντρου.

Στιγμιότυπο για το μικρό δέντρο:

```
Choose one of the above by its number:3
Insert the x value of node to be inserted:GIANNIS
Insert the y value of node to be inserted:1
Insert the z value of node to be inserted:2
Node inserted.
X: Abramsky Y:6 Z:215.0 Range X: Aalst - GIANNIS Range Y: 0 - 10 Range Z: 71.0 - 317.0
-leftchild-X:Abelson Y:5 Z:53.0 Range X: Aalst - Abiteboul Range Y: 0 - 9 Range Z: nan - 317.0
-leftchild--leftchild-X:Abiteboul Y:2 Z:317.0 Range X: Aalst - Abiteboul Range Y: 0 - 2 Range Z: nan - 317.0
-leftchild--leftchild--leftchild-X:Aalst Y:0 Z:nan Range X: Aalst - Aalst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--rightchild-X:Aaronson Y:7 Z:230.0 Range X: Aaronson - Abebe Range Y: 7 - 9 Range Z: 41.0 - 230.0
-leftchild--rightchild--leftchild-X:Abebe Y:9 Z:41.0 Range X: Abebe - Abebe Range Y: 9 - 9 Range Z: 41.0 - 41.0
-rightchild-X:Agrawal Y:9 Z:105.0 Range X: Adleman - GIANNIS Range Y: 1 - 10 Range Z: 1.0 - 105.0
-rightchild--leftchild-X:Adleman Y:2 Z:54.0 Range X: Adleman - GIANNIS Range Y: 1 - 2 Range Z: 1.0 - 54.0
-rightchild--leftchild--leftchild-X:Ahn Y:2 Z:40.0 Range X: GIANNIS - GIANNIS Range Y: 1 - 1 Range Z: 1.0 - 1.0
-rightchild--leftchild--leftchild--rightchild-X:GIANNIS Y:1 Z:1.0 Range X: GIANNIS - GIANNIS Range Y: 1 - 1 Range Z: 1.0 - 1.0
-rightchild--rightchild-X:Atta Y:10 Z:71.0 Range X: Agrawal - Atta Range Y: 9 - 10 Range Z: 71.0 - 105.0

Node inserted in: 0.000627000000014875 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
?) Query search
Choose one of the above by its number:
Insert the x value of node to be inserted:GIANNIS
Insert the y value of node to be inserted:3
Insert the z value of node to be inserted:2
Node inserted.
X: Adleman Y:0 Z:251.0 Range X: Aalst - Zdonik Range Y: 0 - 43 Range Z: 0.0 - 993.0
-leftchild-X:Admeler Y:0 Z:319.0 Range X: Aalst - Kurtz Range Y: 0 - 38 Range Z: nan - 993.0
-leftchild--leftchild-X:Cohen Y:0 Z:85.0 Range X: Aalst - Hehner Range Y: 0 - 0 Range Z: nan - nan
-leftchild--leftchild--leftchild-X:Coldwater Y:0 Z:0.0 Range X: Aalst - Fulkerson Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild-X:Botvinnik Y:0 Z:6.0 Range X: Aalst - Codd Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Barn Y:0 Z:18.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Bezos Y:0 Z:1.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 14.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Asprey Y:0 Z:0.0 Range X: Aalst - Berdichevsky Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Angie Y:0 Z:1.0 Range X: Aalst - Angie Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Zorn Y:0 Z:nan Range X: Aalst - Aalst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Berndichesky Y:0 Z:1.0 Range X: Afre - Berdichevsky Range Y: 0 - 0 Range Z: 0.0 - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Bennicelli Y:0 Z:0.0 Range X: Afre - Afre Range Y: 0 - 0 Range Z: 0.0 - 0.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Attnosoff Y:0 Z:46.0 Range X: Afre - Backhouse Range Y: 0 - 0 Range Z: 0.0 - 46.0
-leftchild--leftchild--leftchild--leftchild--rightchild-X:Bonwick Y:0 Z:7.0 Range X: Blaauw - Booth Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Blaauw Y:0 Z:14.0 Range X: Blaauw - Blum Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Blum Y:0 Z:1.0 Range X: Blum - Blum Range Y: 0 - 0 Range Z: 1.0 - 1.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Booth Y:0 Z:3.0 Range X: Booth - Booth Range Y: 0 - 0 Range Z: 3.0 - 3.0
-leftchild--leftchild--leftchild--leftchild--rightchild-X:Berns Y:0 Z:73.0 Range X: Aragon - Booth Range Y: 0 - 0 Range Z: 23.0 - 83.0
-leftchild--leftchild--leftchild--leftchild--rightchild-X:Atanasoff Y:0 Z:46.0 Range X: Aragon - Backhouse Range Y: 0 - 0 Range Z: 34.0 - 83.0
-leftchild--leftchild--leftchild--leftchild--rightchild-X:Aragon Y:0 Z:83.0 Range X: Aragon - Arden Range Y: 0 - 0 Range Z: 37.0 - 83.0
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Arden Y:0 Z:37.0 Range X: Arden - Arden Range Y: 0 - 0 Range Z: 37.0 - 37.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Backhouse Y:0 Z:71.0 Range X: Babage - Backhouse Range Y: 0 - 0 Range Z: 34.0 - 71.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Backhouse Y:0 Z:34.0 Range X: Babage - Babage Range Y: 0 - 0 Range Z: 34.0 - 34.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Ballacker Y:0 Z:24.0 Range X: Bhushan - Booth Range Y: 0 - 0 Range Z: 23.0 - 50.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Bhushan Y:0 Z:32.0 Range X: Bhushan - Blum Range Y: 0 - 0 Range Z: 23.0 - 32.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Blum Y:0 Z:23.0 Range X: Blum - Blum Range Y: 0 - 0 Range Z: 23.0 - 23.0
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Blum Y:0 Z:50.0 Range X: Booth - Booth Range Y: 0 - 0 Range Z: 50.0 - 50.0
```

```

-rightchild--rightchild--rightchild--leftchild--rightchild-X:halmann Y:5 Z:734.0 Range X: Sihlari - Willmette Range Y: 3 - 5 Range Z: 192.0 - 777.0
-rightchild--rightchild--rightchild--leftchild--rightchild-X:Tamassia Y:6 Z:307.0 Range X: Seifert - Thielmann Range Y: 3 - 4 Range Z: 152.0 - 593.0
-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Thalmann Y:3 Z:593.0 Range X: Sihlari - Thielmann Range Y: 3 - 4 Range Z: 282.0 - 593.0
-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Sihlari Y:4 Z:282.0 Range X: Sihlari - Sigris Range Y: 4 - 4 Range Z: 282.0 - 282.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Tardos Y:3 Z:154.0 Range X: Verdi - Whittaker Range Y: 3 - 3 Range Z: 152.0 - 152.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Tardos Y:4 Z:152.0 Range X: Tardos - Tardos Range Y: 4 - 4 Range Z: 152.0 - 152.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Vardi Y:3 Z:777.0 Range X: Vardi - Nadler Range Y: 3 - 3 Range Z: 169.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wadler Y:3 Z:169.0 Range X: Nadler - Nadler Range Y: 3 - 3 Range Z: 169.0 - 169.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Whittaker Y:5 Z:208.0 Range X: Whittaker - Whittaker Range Y: 5 - 5 Range Z: 208.0 - 208.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Toussaint Y:11 Z:213.0 Range X: Shadforth - Zadeh Range Y: 6 - 26 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Vapnik Y:6 Z:86.0 Range X: Shannon - Wozniak Range Y: 6 - 26 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Sutherland Y:12 Z:52.0 Range X: Shannon - Taylor Range Y: 7 - 26 Range Z: 15.0 - 119.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Taylor Y:7 Z:119.0 Range X: Tate - Taylor Range Y: 7 - 7 Range Z: 63.0 - 119.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Tate Y:7 Z:63.0 Range X: Tate - Tate Range Y: 7 - 7 Range Z: 63.0 - 63.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Shannon Y:15 Z:43.0 Range X: Shannon - Stallman Range Y: 15 - 26 Range Z: 15.0 - 43.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Stallman Y:26 Z:15.0 Range X: Stallman - Stallman Range Y: 26 - 26 Range Z: 15.0 - 15.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wozniak Y:10 Z:13.0 Range X: Vaughan - Wozniak Range Y: 7 - 19 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wiener Y:9 Z:34.0 Range X: Vaughan - Wiener Range Y: 7 - 9 Range Z: 0.0 - 34.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Vaughan Y:7 Z:211.0 Range X: Vaughan - Vaughan Range Y: 7 - 7 Range Z: 0.0 - 0.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wilks Y:19 Z:211.0 Range X: Wilks - Wilks Range Y: 19 - 19 Range Z: 211.0 - 211.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Tanenbaum Y:22 Z:219.0 Range X: Shadforth - Zadeh Range Y: 10 - 15 Range Z: 219.0 - 748.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Shadforth Y:11 Z:294.0 Range X: Shadforth - Sifakis Range Y: 10 - 15 Range Z: 242.0 - 514.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Shadforth Y:10 Z:358.0 Range X: Shadforth - Sifakis Range Y: 10 - 10 Range Z: 242.0 - 338.0
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Sifakis Y:10 Z:242.0 Range X: Sifakis - Sifakis Range Y: 10 - 10 Range Z: 242.0 - 242.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Shneiderman Y:15 Z:514.0 Range X: Schneiderman - Schneiderman Range Y: 15 - 15 Range Z: 514.0 - 514.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yung Y:12 Z:748.0 Range X: Tarjan - Zadeh Range Y: 8 - 17 Range Z: 237.0 - 748.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yung Y:9 Z:627.0 Range X: Tarjan - Veloso Range Y: 8 - 9 Range Z: 350.0 - 627.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Veloso Y:8 Z:350.0 Range X: Tarjan - Tarjan Range Y: 8 - 8 Range Z: 350.0 - 350.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range X: Zadeh - Zadeh Range Y: 17 - 17 Range Z: 237.0 - 237.0

Node inserted in: 0.000276899999889274 seconds

```

Ενημέρωση κόμβου: Η ενημέρωση ενός κόμβου γίνεται μέσω της συνάρτησης update_kd_tree() η οποία δέχεται ως ορίσματα τη ρίζα ενός δέντρου, τον κόμβο προς ενημέρωση, τις νέες τιμές του κόμβου, τη διάσταση του δέντρου και το βάθος της ρίζας. Η συνάρτηση ουσιαστικά πραγματοποιείται καλώντας άλλες συναρτήσεις δηλαδή την delete() για να διαγράψει τον προηγούμενο κόμβο και την insertKdTree() για να εισάγει τον νέο κόμβο. Τέλος μετά την ενημέρωση εκτελείται η συνάρτηση update_range() (εξηγείται παραπάνω) σε ολόκληρο το δέντρο ώστε η τελική του μορφή να είναι η σωστή.

Στιγμιότυπο για το μικρό δέντρο:

```

Choose one of the above by its number:0
Update a node:
Enter old X coordinate: GIANNIS
Enter old Y coordinate: 1
Enter old Z coordinate: 1
Enter new X coordinate: SKAGKOS
Enter new Y coordinate: 2
Enter new Z coordinate: 2
Node updated in: 0.00022200000000793807 seconds
X: Abramsky Y:6 Z:215.0 Range X: Aalst - SKAGKOS Range Y: 0 - 10 Range Z: 71.0 - 317.0
-leftchild-X:Aelson Y:5 Z:53.0 Range X: Aalst - Abiteboul Range Y: 0 - 9 Range Z: nan - 317.0
-leftchild--leftchild-X:Abiteboul Y:2 Z:317.0 Range X: Aalst - Abiteboul Range Y: 0 - 2 Range Z: nan - 317.0
-leftchild--leftchild--leftchild-X:Aalst Y:0 Z:nan Range X: Aalst - Aalst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--rightchild-X:Aaronson Y:7 Z:230.0 Range X: Aaronson - Abebe Range Y: 7 - 9 Range Z: 41.0 - 230.0
-leftchild--rightchild--leftchild-X:Abebe Y:9 Z:41.0 Range X: Abebe - Abebe Range Y: 9 - 9 Range Z: 41.0 - 41.0
-rightchild-X:SKAGKOS Y:2 Z:2.0 Range X: Adleman - SKAGKOS Range Y: 2 - 10 Range Z: 2.0 - 105.0
-rightchild--leftchild-X:Adleman Y:2 Z:54.0 Range X: Adleman - Ahn Range Y: 2 - 2 Range Z: 40.0 - 54.0
-rightchild--leftchild--leftchild-X:Ahn Y:2 Z:40.0 Range X: Ahn - Ahn Range Y: 2 - 2 Range Z: 40.0 - 40.0
-rightchild--rightchild-X:Agrawal Y:9 Z:105.0 Range X: Agrawal - Atta Range Y: 9 - 10 Range Z: 71.0 - 105.0
-rightchild--rightchild--leftchild-X:Atta Y:10 Z:71.0 Range X: Atta - Atta Range Y: 10 - 10 Range Z: 71.0 - 71.0
Give a number for each choice below:

```

Στιγμιότυπο για το μεγάλο δέντρο:

```

Give a number for each choice below:
0) End the program
1) Create the K-Dimensional tree
2) Print the K-Dimensional tree
3) Insert a node
4) Search for a node
5) Delete a node
6) Update a node
7) Query search
Choose one of the above by its number: 6
Update a node:
Enter old X coordinate: 0.0000000000000000
Enter old Y coordinate: 0.0000000000000000
Enter old Z coordinate: 0.0000000000000000
Enter new X coordinate: 0.0000000000000000
Enter new Y coordinate: 0.0000000000000000
Enter new Z coordinate: 0.0000000000000000
Node updated in: 0.009578199999964454 seconds
X:Ladner Y:0 Z:251.0 Range X: Aalst - Zdonik Range Y: 0 - 43 Range Z: 0.0 - 993.0
-leftchild--leftchild-X:Hendler Y:0 Z:319.0 Range X: Aalst - Kolling Range Y: 0 - 38 Range Z: nan - 993.0
-leftchild--leftchild-X:Cohen Y:0 Z:85.0 Range X: Aalst - Neher Range Y: 0 - 0 Range Z: nan - nan
-leftchild--leftchild--leftchild-X:Botvinnik Y:0 Z:6.0 Range X: Aalst - Fulkerson Range Y: 0 - 0 Range Z: nan - 85.0
-leftchild--leftchild--leftchild--leftchild-X:Aalst - Codd Range Y: 0 - 0 Range Z: nan - 6.0
-leftchild--leftchild--leftchild--leftchild-X:Barry Y:0 Z:18.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild-X:Bezos Y:0 Z:1.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 14.0
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Asprey Y:0 Z:0.0 Range X: Aalst - Berdichevsky Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Angle Y:0 Z:1.0 Range X: Aalst - Angie Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild-X:Aalst - Aelst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Berdichevsky Y:0 Z:1.0 Range X: Aelst - Berdichevsky Range Y: 0 - 0 Range Z: 0.0 - 1.0
-leftchild--leftchild--leftchild--leftchild-X:Arte Y:0 Z:0.0 Range X: Arte - Arte Range Y: 0 - 0 Range Z: 0.0 - 0.0
-leftchild--leftchild--leftchild--leftchild-X:Blauw - Booth Range Y: 0 - 0 Range Z: 1.0 - 14.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Vilkinson Y:5 Z:47.0 Range X: Blundin - Wilkins Range Y: 5 - 5 Range Z: 47.0 - 47.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Worley Y:5 Z:1.0 Range X: Worley - Worley Range Y: 5 - 5 Range Z: 1.0 - 1.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Thalmann Y:5 Z:357.0 Range X: Srikar - Whitaker Range Y: 3 - 5 Range Z: 352.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Tamassia Y:4 Z:307.0 Range X: Srikar - Thalmann Range Y: 3 - 4 Range Z: 352.0 - 593.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Talman Y:3 Z:593.0 Range X: Srikar - Thalmann Range Y: 3 - 4 Range Z: 282.0 - 593.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Tardos Y:4 Z:282.0 Range X: Srikar - Srikar Range Y: 4 - 4 Range Z: 282.0 - 282.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Tardos Y:4 Z:152.0 Range X: Tardos - Tardos Range Y: 4 - 4 Range Z: 152.0 - 152.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Varadi - Wagner Y:3 Z:154.0 Range X: Varadi - Wagner Range Y: 3 - 5 Range Z: 154.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Whittaker Y:3 Z:777.0 Range X: Varadi - Wagner Range Y: 3 - 3 Range Z: 169.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Walter Y:3 Z:169.0 Range X: Walter - Walter Range Y: 3 - 3 Range Z: 169.0 - 169.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Whittaker Y:3 Z:208.0 Range X: Whittaker - Whittaker Range Y: 3 - 5 Range Z: 208.0 - 208.0
-rightchild--rightchild--rightchild--rightchild-X:Toussaint Y:11 Z:215.0 Range X: Shabot - Zaden Range Y: 6 - 26 Range Z: 0.0 - 748.0
-rightchild--rightchild--rightchild--rightchild-X:Vapnik Y:6 Z:86.0 Range X: Shannon - Moznik Range Y: 6 - 26 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild-X:Sutherland Y:12 Z:52.0 Range X: Shannon - Taylor Range Y: 7 - 26 Range Z: 15.0 - 119.0
-rightchild--rightchild--rightchild--rightchild-X:Taylor Y:7 Z:119.0 Range X: Tate - Taylor Range Y: 7 - 7 Range Z: 63.0 - 119.0
-rightchild--rightchild--rightchild--rightchild-X:Tate Y:7 Z:63.0 Range X: Tate - Tate Range Y: 7 - 7 Range Z: 63.0 - 63.0
-rightchild--rightchild--rightchild--rightchild-X:Shannon Y:15 Z:43.0 Range X: Shannon - Stallman Range Y: 15 - 26 Range Z: 15.0 - 43.0
-rightchild--rightchild--rightchild--rightchild-X:Stallman Y:26 Z:15.0 Range X: Stallman - Stallman Range Y: 26 - 26 Range Z: 15.0 - 15.0
-rightchild--rightchild--rightchild--rightchild-X:Moznik Y:14 Z:13.0 Range X: Vaughan - Moznik Range Y: 7 - 19 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild-X:Vaughan Y:9 Z:34.0 Range X: Vaughan - Wiener Range Y: 7 - 9 Range Z: 0.0 - 34.0
-rightchild--rightchild--rightchild--rightchild-X:Wiener Y:9 Z:77.0 Range X: Vaughan - Vaughan Range Y: 7 - 7 Range Z: 0.0 - 0.0
-rightchild--rightchild--rightchild--rightchild-X:Wilkes Y:19 Z:211.0 Range X: Wilkes - Wilkes Range Y: 19 - 19 Range Z: 211.0 - 211.0
-rightchild--rightchild--rightchild--rightchild-X:Ramenbaum Y:22 Z:219.0 Range X: Shabot - Zaden Range Y: 6 - 22 Range Z: 219.0 - 748.0
-rightchild--rightchild--rightchild--rightchild-X:Shabot Y:11 Z:294.0 Range X: Shabot - Sifakis Range Y: 10 - 15 Range Z: 242.0 - 514.0
-rightchild--rightchild--rightchild--rightchild-X:Shabot Y:11 Z:338.0 Range X: Shabot - Sifakis Range Y: 10 - 11 Range Z: 242.0 - 338.0
-rightchild--rightchild--rightchild--rightchild-X:Sifakis Y:10 Z:242.0 Range X: Sifakis - Sifakis Range Y: 10 - 10 Range Z: 242.0 - 242.0
-rightchild--rightchild--rightchild--rightchild-X:Shneiderman Y:15 Z:514.0 Range X: Shneiderman - Shneiderman Range Y: 15 - 15 Range Z: 514.0 - 514.0
-rightchild--rightchild--rightchild--rightchild-X:Yuan Y:12 Z:748.0 Range X: Tarjan - Zadeh Range Y: 8 - 17 Range Z: 237.0 - 748.0
-rightchild--rightchild--rightchild--rightchild-X:Veloso Y:9 Z:627.0 Range X: Tarjan - Veloso Range Y: 8 - 9 Range Z: 350.0 - 627.0
-rightchild--rightchild--rightchild--rightchild-X:Tarjan Y:8 Z:350.0 Range X: Tarjan - Tarjan Range Y: 8 - 8 Range Z: 350.0 - 350.0
-rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range X: Zadeh - Zadeh Range Y: 17 - 17 Range Z: 237.0 - 237.0

```

Διαγραφή κόμβου: Η λειτουργία διαγραφής ενός κόμβου υλοποιείται με την συνάρτηση `delete_kd()` η οποία δέχεται ως ορίσματα την ρίζα του δέντρου, τον κόμβο προς διαγραφή και το βάθος της ρίζας. Αρχικά το δέντρο χωρίζεται κατά διαφορετικές διαστάσεις (x, y, z) με κυκλικό τρόπο βάσει του τρέχοντος βάθους (`current_dim = depth % 3`). Η συνάρτηση ελέγχει εάν ο τρέχων κόμβος έχει δεξί υποδέντρο και εάν έχει βρίσκει τον ελάχιστο κόμβο στο δεξί υποδέντρο κατά μήκος της τρέχουσας διάστασης, αντικαθιστά τις συντεταγμένες του τρέχοντος κόμβου με αυτές του ελάχιστου κόμβου, και στη συνέχεια διαγράφει αναδρομικά τον ελάχιστο κόμβο. Εάν δεν έχει δεξί υποδέντρο, η συνάρτηση επιστρέφει το αριστερό υποδέντρο (αφαιρώντας τον τρέχοντα κόμβο). Εάν ο κόμβος που πρόκειται να διαγραφεί δεν βρεθεί, η συνάρτηση συνεχίζει την αναζήτηση στο κατάλληλο υποδέντρο με βάση τη σύγκριση των συντεταγμένων. Η συνάρτηση `find_min()` δέχεται τα ίδια ορίσματα με την `delete_kd()` και χρησιμοποιείται για να βρίσκει και επιστρέφει τον κόμβο με την ελάχιστη τιμή κατά μήκος μιας συγκεκριμένης διάστασης σε ένα υποδέντρο.

Στιγμιότυπο για το μικρό δέντρο:

```
Choose one of the above by its number:5
Insert the x value of node to be deleted:SKAKKOS
Insert the y value of node to be deleted:2
Insert the z value of node to be searched:2
Node updated in: 2.740000000454838e-05 seconds
Tree after deleting the node:
X: Abramsky Y:6 Z:215.0 Range X: Aalst - Atta Range Y: 0 - 10 Range Z: 71.0 - 317.0
-leftchild-X:Abelson Y:5 Z:53.0 Range X: Aalst - Abiteboul Range Y: 0 - 9 Range Z: nan - 317.0
-leftchild--leftchild-X:Abiteboul Y:2 Z:317.0 Range X: Aalst - Abiteboul Range Y: 0 - 2 Range Z: nan - 317.0
-leftchild--leftchild--leftchild-X:Aalst Y:0 Z:nan Range X: Aalst - Aalst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--rightchild-X:Aaronson Y:7 Z:230.0 Range X: Aaronson - Abebe Range Y: 7 - 9 Range Z: 41.0 - 230.0
-leftchild--rightchild--leftchild-X:Abebe Y:9 Z:41.0 Range X: Abebe - Abebe Range Y: 9 - 9 Range Z: 41.0 - 41.0
-rightchild-X:Agrawal Y:9 Z:105.0 Range X: Adleman - Atta Range Y: 2 - 10 Range Z: 40.0 - 105.0
-rightchild--leftchild-X:Adleman Y:2 Z:54.0 Range X: Adleman - Ahn Range Y: 2 - 2 Range Z: 40.0 - 54.0
-rightchild--leftchild--leftchild-X:Ahn Y:2 Z:40.0 Range X: Ahn - Ahn Range Y: 2 - 2 Range Z: 40.0 - 40.0
-rightchild--rightchild-X:Atta Y:10 Z:71.0 Range X: Atta - Atta Range Y: 10 - 10 Range Z: 71.0 - 71.0
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
Give a number for each choice below:
0) End the program
1) Create the K-Dimensional tree
2) Print the K-Dimensional tree
3) Insert a node
4) Search for a node
5) Delete a node
6) Update a node
7) Query search
Choose one of the above by its number:5
Insert the x value of node to be deleted: XACOOS
Insert the y value of node to be deleted:
Insert the z value of node to be searched:
Node updated in: 9.189999915767e-05 seconds
Tree after deleting the node:
X:Ladner Y:0 Z:251.0 Range X: Aalst - Zdonik Range Y: 0 - 43 Range Z: 0.0 - 993.0
-leftchild-X:Hendler Y:0 Z:319.0 Range X: Aalst - Kolling Range Y: 0 - 38 Range Z: nan - 993.0
-leftchild--leftchild-X:Cohen Y:0 Z:85.0 Range X: Aalst - Hennar Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild-X:Goldwater Y:0 Z:0.0 Range X: Aalst - Fulkeron Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild-X:Botvinnik Y:0 Z:6.0 Range X: Aalst - Codd Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild-X:Barn Y:0 Z:18.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 83.0
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Bezos Y:0 Z:1.0 Range X: Aalst - Booth Range Y: 0 - 0 Range Z: nan - 14.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Asprey Y:0 Z:9.0 Range X: Aalst - Berdichevsky Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Angie Y:0 Z:1.0 Range X: Aalst - Angle Range Y: 0 - 0 Range Z: nan - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Alast Y:0 Z:nan Range X: Alast - Aalst Range Y: 0 - 0 Range Z: nan - nan
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Berdichevsky Y:0 Z:1.0 Range X: Atre - Berdichevsky Range Y: 0 - 0 Range Z: 0.0 - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Atre Y:0 Z:0.0 Range X: Atre - Atre Range Y: 0 - 0 Range Z: 0.0 - 0.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Bonwick Y:0 Z:7.0 Range X: Blaauw - Booth Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Bonwick Y:0 Z:14.0 Range X: Blaauw - Blum Range Y: 0 - 0 Range Z: 1.0 - 14.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Blaauw Y:0 Z:1.0 Range X: Blum - Blum Range Y: 0 - 0 Range Z: 1.0 - 1.0
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Booth Y:0 Z:33.0 Range X: Booth - Booth Range Y: 0 - 0 Range Z: 3.0 - 3.0
-leftchild--leftchild--leftchild--rightchild-X:Berms Y:0 Z:73.0 Range X: Aragon - Booth Range Y: 0 - 0 Range Z: 25.0 - 83.0
-rightchild-X:Richter Y:0 Z:152.0 Range X: Wilminson Y:5 Z:14.0 Range X: Wilminson - Wilminson Range Y: 5 - 5 Range Z: 47.0 - 47.0
-rightchild--rightchild--rightchild-X:Worley Y:5 Z:1.0 Range X: Worley - Worley Range Y: 5 - 5 Range Z: 1.0 - 1.0
-rightchild--rightchild--rightchild--rightchild-X:Thallmann Y:5 Z:734.0 Range X: Spihari - Whittaker Range Y: 0 - 5 Range Z: 152.0 - 777.0
-rightchild--rightchild--rightchild--rightchild-X:Tamassia Y:4 Z:307.0 Range X: Srihari - Thamani Range Y: 3 - 4 Range Z: 152.0 - 593.0
-rightchild--rightchild--rightchild--rightchild-X:Thamani Y:3 Z:593.0 Range X: Srihari - Thamani Range Y: 3 - 4 Range Z: 282.0 - 593.0
-rightchild--rightchild--rightchild--rightchild-X:Thalhaan Y:4 Z:152.0 Range X: Srihari - Srihari Range Y: 4 - 4 Range Z: 282.0 - 282.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Thalhaan Y:4 Z:152.0 Range X: Tardos - Tardos Range Y: 4 - 4 Range Z: 152.0 - 152.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wegner Y:5 Z:154.0 Range X: Vardi - Whittaker Range Y: 3 - 5 Range Z: 152.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Vardi Y:3 Z:777.0 Range X: Vardi - Vardi Range Y: 3 - 5 Range Z: 169.0 - 777.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Whittaker Y:3 Z:169.0 Range X: Whittaker - Whittaker Range Y: 3 - 3 Range Z: 169.0 - 169.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Whittaker Y:5 Z:208.0 Range X: Whittaker - Whittaker Range Y: 5 - 5 Range Z: 208.0 - 208.0
-rightchild--rightchild--rightchild--rightchild-X:Toussaint Y:11 Z:213.0 Range X: Shannon - Zhen Range Y: 0 - 20 Range Z: 0.0 - 748.0
-rightchild--rightchild--rightchild--rightchild-X:Shannon Y:10 Z:86.0 Range X: Shannon - Novikov Range Y: 0 - 26 Range Z: 0.0 - 221.0
-rightchild--rightchild--rightchild--rightchild-X:Novikov Y:10 Z:252.0 Range X: Shannon - Taylor Range Y: 0 - 26 Range Z: 15.0 - 119.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Sunderland Y:12 Z:52.0 Range X: Shannon - Taylor Range Y: 7 - 26 Range Z: 15.0 - 119.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Taylor Y:7 Z:119.0 Range X: Take - Take Range Y: 7 - 7 Range Z: 63.0 - 119.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Take Y:7 Z:63.0 Range X: Take - Take Range Y: 7 - 7 Range Z: 63.0 - 63.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Shannon Y:15 Z:45.0 Range X: Shannon - Stallman Range Y: 15 - 26 Range Z: 15.0 - 15.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Stallman Y:26 Z:15.0 Range X: Stallman - Stallman Range Y: 26 - 26 Range Z: 15.0 - 15.0
-rightchild--rightchild--rightchild--rightchild-X:Wozniak Y:14 Z:13.0 Range X: Vaagen - Wozniak Range Y: 7 - 19 Range Z: 0.0 - 211.0
-rightchild--rightchild--rightchild--rightchild-X:Vaagen Y:9 Z:34.0 Range X: Vaagen - Wiener Range Y: 7 - 9 Range Z: 0.0 - 34.0
-rightchild--rightchild--rightchild--rightchild--rightchild-X:Wiener Y:9 Z:34.0 Range X: Vaagen - Wiener Range Y: 7 - 9 Range Z: 0.0 - 34.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Vaughn Y:7 Z:0.0 Range X: Vaughn - Vaughn Range Y: 7 - 7 Range Z: 0.0 - 0.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Wilks Y:19 Z:211.0 Range X: Wilks - Wilks Range Y: 19 - 19 Range Z: 211.0 - 211.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Tanenbaum Y:21 Z:219.0 Range X: Shamot - Zadeh Range Y: 8 - 22 Range Z: 219.0 - 748.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Shamot Y:11 Z:294.0 Range X: Shamot - Sifakis Range Y: 10 - 15 Range Z: 242.0 - 514.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Sifakis Y:11 Z:338.0 Range X: Shamot - Sifakis Range Y: 10 - 11 Range Z: 242.0 - 338.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Sifakis Y:10 Z:242.0 Range X: Shamot - Sifakis Range Y: 10 - 10 Range Z: 242.0 - 242.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Shneiderman Y:15 Z:514.0 Range X: Shneiderman - Shneiderman Range Y: 15 - 15 Range Z: 514.0 - 514.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yann Y:12 Z:748.0 Range X: Tarijan - Tarijan Range Y: 8 - 17 Range Z: 237.0 - 748.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Veloso Y:9 Z:627.0 Range X: Tarijan - Veloso Range Y: 8 - 9 Range Z: 350.0 - 627.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Tarijan Y:8 Z:350.0 Range X: Tarijan - Tarijan Range Y: 8 - 8 Range Z: 350.0 - 350.0
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range X: Zadeh - Zadeh Range Y: 17 - 17 Range Z: 237.0 - 237.0
```

Αναζήτηση εύρους με Ish μέθοδο: Η συνάρτηση `query_range()` παίρνει τη ρίζα του δέντρου εύρους 3D και διάφορες παραμέτρους ερωτήματος όπως `min_letter`, `max_letter`, `min_y`, `max_y`, `min_z` και `max_z`. Ελέγχει εάν ο τρέχων κόμβος (ρίζα) ικανοποιεί τα κριτήρια του ερωτήματος με βάση τις συνθήκες που παρέχονται. Οι συνθήκες περιλαμβάνουν τον έλεγχο εάν η συντεταγμένη x εμπίπτει στο

καθορισμένο εύρος γραμμάτων (`min_gamma` μέχρι `max_gamma`), εάν η συντεταγμένη γ είναι εντός του καθορισμένου εύρους (`min_y` μέχρι `max_y`) και εάν η συντεταγμένη z είναι εντός του καθορισμένου εύρους (`min_z` μέχρι `max_z`). Εάν πληρούνται οι προϋποθέσεις, η συνάρτηση ανακτά πληροφορίες εκπαίδευσης από το αρχικό σύνολο δεδομένων με βάση το επώνυμο (`root.x`) και προσαρτά ένα record που περιέχει τις συντεταγμένες και τις πληροφορίες εκπαίδευσης του κόμβου στη λίστα αποτελεσμάτων. Στη συνέχεια, η συνάρτηση καλείται αναδρομικά στο αριστερό και το δεξί υποδέντρο με βάση τη συντεταγμένη x και, εάν ισχύει, στο σχετικό δέντρο Y. Η αναδρομή συνεχίζεται έως ότου γίνει επίσκεψη όλων των κόμβων που πληρούν τα κριτήρια του ερωτήματος. Η λίστα αποτελεσμάτων τροποποιείται κατά την εκτέλεση της συνάρτησης και θα περιέχει πληροφορίες για όλους τους κόμβους που ικανοποιούν τις συνθήκες του ερωτήματος. Μετά την εκτέλεση της συνάρτησης εκτελούνται συγκεκριμένα βήματα που περιλαμβάνουν την εξαγωγή εκπαιδευτικών πληροφοριών από τα αποτελέσματα του ερωτήματος, την εφαρμογή Locality-Sensitive Hashing (LSH) για την εύρεση παρόμοιων εγγραφών με βάση τις πληροφορίες εκπαίδευσης τους και την εκτύπωση των παρόμοιων εγγραφών εκπαίδευσης.

Στιγμιότυπο για το μικρό δέντρο:

```
Choose one of the above by its number:  
Enter the minimum letter limit from A to Z, a to z: 1  
Enter the maximum letter limit from A to Z, a to z: 2  
Enter the minimum awards value: 0  
Enter the minimum DBLRecord value: 0  
Enter the maximum DBLRecord value: 100  
Enter the similarity threshold (0 to 100): 10  
["'Abramsky', 6, 215.0, "'Abramsky was educated at Hasmonean Grammar School for Boys, Hendon and at King's College, Cambridge (BA 1975, MA Philosophy 1979, Diploma in Computer Science) and Queen Mary, University of London (PhD 1983). He graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He has been a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology."],  
Query Results:  
("'Abramsky', 6, 215.0, "'Abramsky was educated at Hasmonean Grammar School for Boys, Hendon and at King's College, Cambridge (BA 1975, MA Philosophy 1979, Diploma in Computer Science) and Queen Mary, University of London (PhD 1983). He graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He is a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology.'"),  
("'Abelson', 5, 53.0, "'Abelson graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He is a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology.'"),  
("'Aaronsen', 7, 230.0, "nan"),  
("'Aabebe', 9, 41.0, "nan"),  
("'Agrawal', 9, 105.0, "nan"),  
("'Adleman', 2, 54.0, "nan"),  
("'Ahn', 2, 40.0, "nan"),  
("'Atta', 10, 71.0, "'Khan was a Bright Sparks scholar and received his PhD degree in Computer Science from University of Malaya.[14] He received his master's degree and bachelor's degree (with honors) in Computer Science from the University of Malaya. He graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He is a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology.'"),  
Similar Education Records:  
("'Aabebe', 9, 41.0, "nan"),  
("'Abramsky', 6, 215.0, "'Abramsky was educated at Hasmonean Grammar School for Boys, Hendon and at King's College, Cambridge (BA 1975, MA Philosophy 1979, Diploma in Computer Science) and Queen Mary, University of London (PhD 1983). He graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He is a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology.'"),  
("'Adleman', 2, 54.0, "nan"),  
("'Aaronsen', 7, 230.0, "nan"),  
("'Ahn', 2, 40.0, "nan"),  
("'Agrawal', 9, 105.0, "nan"),  
("'Atta', 10, 71.0, "'Khan was a Bright Sparks scholar and received his PhD degree in Computer Science from University of Malaya.[14] He received his master's degree and bachelor's degree (with honors) in Computer Science from the University of Malaya. He graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point sets: a homology sphere, and a Master of Arts degree in philosophy from Princeton in 1971. He is a professor of computer science at the University of Edinburgh since 1990. He is a fellow of the Royal Society of Edinburgh and a member of the Royal Society of London. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology. He is a member of the International Association for Symbolic Mathematics and the European Association for Computer Science and Technology.'"),  
None found in: 0.0721820000000056 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
1) Delete a node  
2) Update a node  
3) Insert a node  
4) Query search  
Choose one of the above by its number:  
Enter the minimum letter limit from A to Z, a to z:  
Enter the maximum letter limit from A to Z, a to z:  
Enter the minimum awards value:  
Enter the minimum DBLPRecord value:  
Enter the maximum DBLPRecord value:  
Enter the similarity threshold @ to 100: 80  
[('Date', 0, 89.0, nan), ('Garis', 0, 95.0, nan), ('Garey', 0, 63.0, nan), ('Gallerter', 0, 60.0, nan), ('Gelernter', 0, 68.0, nan), ('Gates', 0, 18.0, nan), ('Goldberg', 0, 61.0, nan), ('Greibach', 0, 77.0, nan), ('Gruber', 0, 45.0, nan), ('Griswold', 0, 48.0, nan), ('Griesemer', 0, 10.0, nan), ('Gutknecht', 0, 47.0, nan)]  
Similar Education Records:  
[('Gutknecht', 0, 47.0, nan), ('Date', 0, 89.0, nan), ('Galler', 0, 60.0, nan), ('Garis', 0, 95.0, nan), ('Greibach', 0, 77.0, nan), ('Gelernter', 0, 68.0, nan), ('Griswold', 0, 48.0, nan), ('Garey', 0, 63.0, nan), ('Goldberg', 0, 61.0, nan), ('Gruber', 0, 45.0, nan), ('Gates', 0, 18.0, nan), ('Griesemer', 0, 10.0, nan)]
```

Πίνακας Σύγκρισης Αποτελεσμάτων

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
KD Δημιουργία δέντρου	0.0000920000000000399	0.009561500000000223
KD Αναζήτηση κόμβου	0.0000342000000017606	0.00004179999996267725
KD Εισαγωγή κόμβου	0.00062700000000014875	0.006276899999989274
KD Ενημέρωση κόμβου	0.00022200000000793807	0.009578199999964454
KD Διαγραφή κόμβου	0.000027400000000454838	0.0000918999999157677
KD Αναζήτηση εύρους (Ish)	0.07121820000000056	0.0637988999999833

RANGE-TREE: Η παρακάτω υλοποίηση αφορά την κατασκευή και επεξεργασία ενός Range tree 3 διαστάσεων το οποίο περιέχει τα δεδομένα ενός πλήθους επιστημόνων, δηλαδή το όνομα, τον αριθμό των βραβείων και των αριθμό του DBLP ιστορικού τους στις x,y και z διαστάσεις αντίστοιχα, τα οποία αντλούνται από ένα csv αρχείο. Ουσιαστικά δημιουργούνται 3 δέντρα τα xtree, ytree και ztree το καθένα από τα οποία περιέχει τα δεδομένα του αρχείου και τα ταξινομεί με βάσει την αντίστοιχη συντεταγμένη (xtree βάσει του x, ytree βάσει του y και ztree βάσει του z). Ο χρήστης καλείται να επιλέξει ανάμεσα σε 7 λειτουργίες όσον αφορά την επεξεργασία του δέντρου, οι οποίες παρουσιάζονται και αναλύονται παρακάτω. Να σημειωθεί πως η εκτέλεση και τα παραδείγματα γίνονται για 2 αρχεία με δεδομένα, ένα το οποίο περιέχει 10 επιστήμονες σύνολο και άλλο ένα με τους συνολικούς επιστήμονες. Σκοπός αυτού του διαχωρισμού των αρχείων είναι η ευκολότερη επεξεργασία, κατανόηση και υλοποίηση των λειτουργιών, ώστε να δουλεύουν τόσο σε μικρό όσο και σε μεγάλο όγκο δεδομένων.

ΛΕΙΤΟΥΡΓΙΕΣ:

Δημιουργία δέντρου: Αρχικά δημιουργείται η κλάση κόμβου ,η οποία αντιπροσωπεύει έναν κόμβο στο 3D δέντρο. Κάθε κόμβος περιέχει τρεις συντεταγμένες (x, y, z), δείκτες στα αριστερά και δεξιά παιδιά του, δείκτες σε συναφή δέντρα σε άλλες διαστάσεις (το X-δέντρο έχει y-associated tree κ.λπ.) και προαιρετικές τιμές min/max για αποδοτικά ερωτήματα εύρους. Το δέντρο δημιουργείται με την κλήση της makeRange3dTree(),η οποία παίρνει μια λίστα σημείων (το καθένα αντιπροσωπεύεται ως [x, y, z]) και κατασκευάζει αναδρομικά το δέντρο με βάση τις συντεταγμένες σε κάθε διάσταση (x, y, z).

Πιο συγκεκριμένα εάν η λίστα των σημείων είναι κενή, επιστρέφει κενά, καθώς δεν υπάρχουν κόμβοι για δημιουργία. Προσδιορίζει τον διάμεσο δείκτη της ταξινομημένης λίστας σημείων(median). Αυτή θα είναι η ρίζα. Ταξινομεί την λίστα των σημείων με βάση την επιλεγμένη διάσταση (x, y ή z). Δημιουργεί έναν κόμβο για τη ρίζα χρησιμοποιώντας τις συντεταγμένες του μέσου σημείου στη λίστα ταξινόμησης. Ρυθμίζει τις τιμές MINR και MAXR για αποδοτικά ερωτήματα εύρους.

Δημιουργεί αναδρομικά τα αριστερά και τα δεξιά subtrees καλώντας το Makerange3Dtree στα τμήματα της λίστας που ταξινομείται πριν και μετά το διάμεσο και ανάλογα με την επιλεγμένη διάσταση (x, y ή z), η συνάρτηση ταξινομεί και χωρίζει τη λίστα ανάλογα.

Εκτύπωση δέντρου: Η λειτουργία PrintTree είναι μια αναδρομική λειτουργία που διασχίζει το 3D δέντρο και εκτυπώνει πληροφορίες για κάθε κόμβο σε μια

αναγνώσιμη μορφή. Εκτυπώνει λεπτομέρειες σχετικά με τις συντεταγμένες x, y και z κάθε κόμβου, καθώς και το εύρος των τιμών στη διάσταση x (minr σε maxr). Επιπλέον, εκτυπώνει πληροφορίες σχετικά με το σχετικό Y-Tree (Assoc) εάν υπάρχει.

Στιγμιότυπο για το μικρό δέντρο:

```
2
The Range Tree:
X:Anthony Y:6 Z:317 Range: Abelson - Barto
-lefthchild-X:Abromsky Y:6 Z:40 Range: Abelson - Anderson
-lefthchild--lefthchild-X:Abiteboul Y:3 Z:230 Range: Abelson - Abiteboul
-lefthchild--lefthchild--lefthchild-X:Abelson Y:5 Z:116 Range: Abelson - Abelson
-lefthchild--rightchild-X:Anderson Y:0 Z:52 Range: Anderson - Anderson
-rightchild-X:Atta Y:7 Z:71 Range: Aragon - Barto
-rightchild--lefthchild-X:Atre Y:0 Z:54 Range: Aragon - Atre
-rightchild--lefthchild--lefthchild-X:Aragon Y:0 Z:215 Range: Aragon - Aragon
-rightchild--rightchild-X:Barto Y:3 Z:105 Range: Barto - Barto

Create Range Tree executed in: 4.000000023354687e-07 seconds
Give a number for each choice below:
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
7) Query search
9
The Range Tree:
X:Ladner Y:0 Z:251.0 Range: Aalst - Zdonik
-lefthchild-X:Fayed Y:0 Z:3.0 Range: Aalst - KÄ¶lling
-lefthchild--lefthchild-X:Cardelli Y:3 Z:250.0 Range: Aalst - Fahlman
-lefthchild--lefthchild--lefthchild-X:Bhushan Y:0 Z:32.0 Range: Aalst - Cantrill
-lefthchild--lefthchild--lefthchild--lefthchild-X:Atre Y:0 Z:0.0 Range: Aalst - Bezos
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Allen Y:5 Z:37.0 Range: Aalst - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Abromsky Y:6 Z:215.0 Range: Aalst - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Abel Y:9 Z:41.0 Range: Aalst - Abiteboul
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Aaronson Y:7 Z:230.0 Range: Aalst - Aaronson
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Aalst Y:0 Z:1119.0 Range: Aalst - Aalst
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Abiteboul Y:2 Z:317.0 Range: Abelson - Abiteboul
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild-X:Abelson Y:5 Z:53.0 Range: Abelson - Abelson
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Ahm Y:2 Z:40.0 Range: Adleman - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Agrawal Y:9 Z:105.0 Range: Adleman - Agrawal
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild--lefthchild-X:Adleman Y:2 Z:54.0 Range: Adleman - Adleman
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild-X:Alfred Y:0 Z:113.0 Range: Alfred - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Atanasoff Y:0 Z:83.0 Range: Amdahl - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Angie Y:0 Z:1.0 Range: Amdahl - Appel
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Anderson Y:0 Z:573.0 Range: Amdahl - Anderson
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild--lefthchild--lefthchild-X:Andahl Y:4 Z:16.0 Range: Amdahl - Amdahl
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild--rightchild-X:Appel Y:0 Z:148.0 Range: Anthony - Appel
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild--rightchild-X:Anthony Y:6 Z:148.0 Range: Anthony - Anthony
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild-X:Asprey" Y:0 Z:0.0 Range: Arden - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild-X:Arora Y:0 Z:245.0 Range: Arden - Arora
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild--lefthchild-X:Arden Y:0 Z:37.0 Range: Arden - Arden
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild--rightchild-X:Atanasoff Y:0 Z:46.0 Range: Atanasoff - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Barto Y:2 Z:219.0 Range: Atta - Bezos
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Bacon Y:0 Z:124.0 Range: Atta - Bartik
```

```

-rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--leftchild-X:Wirth Y:0 Z:115.0 Range: Wirth - Wirth
-rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild-X:Woodger Y:0 Z:0.0 Range: Wolfram - Woodger
-rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild-X:Wolfra Y:0 Z:56.0 Range: Wolfram - Wolfram
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Wu Y:0 Z:nan Range: Worsley - Wulf
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Wozniak Y:14 Z:13.0 Range: Worsley - Wozniak
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--leftchild-X:Worsley Y:5 Z:1.0 Range: Worsley - Worsley
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--rightchild-X:Wulf Y:0 Z:150.0 Range: Wulf - Wulf
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--rightchild-X:Yung Y:12 Z:748.0 Range: Yao - Zdonik
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yoneda Y:0 Z:1.0 Range: Yao - Yourdon
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yen Y:0 Z:419.0 Range: Yao - Yen
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yao Y:1 Z:179.0 Range: Yao - Yao
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yourdon Y:0 Z:22.0 Range: Yourdon - Yourdon
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zantema Y:0 Z:118.0 Range: Zadeh - Zdonik
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zaman Y:0 Z:35.0 Range: Zadeh - Zaman
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range: Zadeh - Zadeh
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zdonik Y:0 Z:194.0 Range: Zdonik - Zdonik

```

Create Range Tree executed in: 5.0000002290335e-07 seconds

Εισαγωγή κόμβου: Η συνάρτηση insert() λειτουργεί ως εξής: Εάν ο τρέχων κόμβος (root) δεν υπάρχει, η συνάρτηση επιστρέφει χωρίς να εκτελεί καμία εισαγωγή. Ανάλογα με την επιλεγμένη διάσταση (x, y ή z), η συνάρτηση καθορίζει αν θα εισαγάγετε τον νέο κόμβο προς τα αριστερά ή δεξιά υπό -υποκείμενο του τρέχοντος κόμβου. Εάν η συντεταγμένη του νέου κόμβου ταιριάζει με τη συντεταγμένη του τρέχοντος κόμβου στην καθορισμένη διάσταση, η συνάρτηση επιστρέφει χωρίς εισαγωγή .Καλεί αναδρομικά τη λειτουργία εισαγωγής στο κατάλληλο subtree (αριστερά ή δεξιά) με βάση τη σύγκριση των συντεταγμένων. Εάν όχι στο Y-Tree ή στο Z-Tree, εισάγει αναδρομικά τον κόμβο στο σχετικό Y-Tree (Assoc), καλώντας το Insert με τις αντίστοιχες παραμέτρους.

Στιγμιότυπο για το μικρό δέντρο:

```

3
Give the coordinates for the node to insert:
Enter X coordinate: Mario
Enter Y coordinate: 1
Enter Z coordinate: 1
Tree after inserting the node:
X:Aragon Y:0 Z:215 Range: Abelson - Mario
-leftchild-X:Abromsky Y:6 Z:40 Range: Abelson - Anthony
-leftchild--leftchild-X:Abiteboul Y:3 Z:230 Range: Abelson - Abiteboul
-leftchild--leftchild--leftchild-X:Abelson Y:5 Z:116 Range: Abelson - Abelson
-leftchild--rightchild-X:Anthony Y:6 Z:317 Range: Anderson - Anthony
-leftchild--rightchild--leftchild-X:Anderson Y:0 Z:52 Range: Anderson - Anderson
-rightchild-X:Barto Y:3 Z:105 Range: Atre - Mario
-rightchild--leftchild-X:Att-a Y:7 Z:71 Range: Atre - Att-a
-rightchild--leftchild--leftchild-X:Atre Y:0 Z:54 Range: Atre - Atre
-rightchild--rightchild-X:Mario Y:1 Z:1 Range: Mario - Mario

Insert node executed in: 9.899999999340992e-06 seconds

```

Στιγμιότυπο για το μεγάλο δέντρο:

```
3
Give the coordinates for the node to insert:
Enter X coordinate: tom
Enter Y coordinate: 1
Enter Z coordinate: 3
Tree after inserting the node:
X:Ladner Y:0 Z:251.0 Range: Aalst - tom
-leftchild-X:Fayed Y:0 Z:3.0 Range: Aalst - Käffling
-leftchild--leftchild-X:Cardelli Y:3 Z:250.0 Range: Aalst - Fahlman
-leftchild--leftchild--leftchild-X:Bhushan Y:0 Z:32.0 Range: Aalst - Cantrill
-leftchild--leftchild--leftchild-X:Atre Y:0 Z:0.0 Range: Aalst - Bezos
-leftchild--leftchild--leftchild--leftchild-X:Allen Y:5 Z:37.0 Range: Aalst - Atanasoff
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Abraamsky Y:6 Z:215.0 Range: Aalst - Alfred
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Abbe Y:9 Z:41.0 Range: Aalst - Abiteboul
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Aaronson Y:7 Z:230.0 Range: Aalst - Aaronson
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Aalst Y:0 Z:1119.0 Range: Aalst - Aalst
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Abiteboul Y:2 Z:317.0 Range: Abelson - Abiteboul
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Abelson Y:5 Z:53.0 Range: Abelson - Abelson
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Abelson Y:2 Z:54.0 Range: Adleman - Adleman
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Agrawal Y:9 Z:105.0 Range: Adleman - Agrawal
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Aldelean Y:2 Z:54.0 Range: Adleman - Adleman
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Alfred Y:0 Z:113.0 Range: Alfred - Alfred
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Aragon Y:0 Z:83.0 Range: Amdahl - Atanasoff
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Angie Y:0 Z:1.0 Range: Amdahl - Appel
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Anderson Y:0 Z:573.0 Range: Amdahl - Anderson
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild--leftchild-X:Amdahl Y:4 Z:10.0 Range: Amdahl - Amdahl
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild--rightchild-X:Appel Y:0 Z:148.0 Range: Anthony - Appel
-leftchild--leftchild--leftchild--leftchild--rightchild--leftchild--rightchild-X:Anthony Y:6 Z:148.0 Range: Anthony - Anthony
-leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Asprey" Y:0 Z:0.0 Range: Arden - Atanasoff

-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild-X:Wisnrosky Y:0 Z:0.0 Range: Wiseman - Wisnrosky
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--leftchild-X:Wiseman Y:0 Z:29.0 Range: Wiseman - Wiseman
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild-X:Woodward Y:0 Z:20.0 Range: Woodger - Woodward
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild-X:Woodger Y:0 Z:0.0 Range: Woodger - Woodger
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild-X:Wulf Y:0 Z:150.0 Range: Wozniak - Yannakakis
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--leftchild-X:Wu Y:0 Z:nan Range: Wozniak - Wu
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild--rightchild--leftchild--leftchild--leftchild-X:Wozniak Y:14 Z:13.0 Range: Wozniak - Wozniak
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yannakakis Y:1 Z:291.0 Range: Yannakakis - Yannakakis
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range: Yen - tom
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Youndon Y:0 Z:22.0 Range: Yen - Yung
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yoneda Y:0 Z:1.0 Range: Yen - Yoneda
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yen Y:0 Z:419.0 Range: Yen - Yen
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Yung Y:12 Z:748.0 Range: Yung - Yung
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zdonik Y:0 Z:194.0 Range: Zaman - tom
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zantema Y:0 Z:118.0 Range: Zaman - Zantema
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zaman Y:0 Z:35.0 Range: Zaman - Zaman
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:tom Y:1 Z:1.0 Range: tom - tom

Insert node executed in: 1.53999999522479e-05 seconds
```

Διαγραφή κόμβου: Η διαγραφή κόμβου γίνεται με την `delete()` η οποία λειτουργεί ως εξής: Εάν ο τρέχων κόμβος (ρίζα) δεν υπάρχει, η συνάρτηση δεν επιστρέφει καμία, υποδεικνύοντας ότι ο κόμβος που πρόκειται να διαγραφεί δεν βρέθηκε. Ανάλογα με την επιλεγμένη διάσταση (x , y ή z), η συνάρτηση αναζητά τον κόμβο που πρέπει να διαγραφεί στο κατάλληλο subtree. Εάν ο κόμβος που πρόκειται να διαγραφεί βρίσκεται στο X-Tree, η συνάρτηση μεταβαίνει στο Y-Tree και Z-Tree για διαγραφή και καλεί τον εαυτό της αναδρομικά. Εάν ο κόμβος που πρέπει να διαγραφεί βρίσκεται στο Y-Tree, η λειτουργία μεταβαίνει στο X-Tree και Z-Tree για διαγραφή. Εάν ο κόμβος που πρέπει να διαγραφεί βρίσκεται στο Z-Tree, η λειτουργία μεταβαίνει στο X-Tree και στο Y-Tree για διαγραφή.

Εάν ο κόμβος που πρέπει να διαγραφεί βρίσκεται και στις τρεις διαστάσεις, σημαίνει ότι ο κόμβος βρίσκεται στις καθορισμένες συντεταγμένες. Η λειτουργία προχωρά στη διαγραφή του κόμβου με βάση τον αριθμό των παιδιών που έχει. Εάν ο κόμβος έχει δύο παιδιά, η λειτουργία βρίσκει τον ελάχιστο κόμβο στο σωστό subtree. Στη συνέχεια αντιγράφει τις τιμές από τον ελάχιστο κόμβο στον τρέχοντα κόμβο. Τέλος, διαγράφει αναδρομικά τον ελάχιστο κόμβο από το δεξιό subtree και επιστρέφει την τροποποιημένη ρίζα μετά τη διαγραφή.

Στιγμιότυπο για το μικρό δέντρο:

```
4
Give the coordinates for the node to delete:
Enter X coordinate to delete: Mario
Enter Y coordinate to delete: 1
Enter Z coordinate to delete: 1
Tree after deleting the node:
X:Anthony Y:6 Z:317 Range: Abelson - Barto
-lefthchild-X:Abramsky Y:6 Z:40 Range: Abelson - Anderson
-lefthchild--lefthchild-X:Abiteboul Y:3 Z:230 Range: Abelson - Abiteboul
-lefthchild--lefthchild--lefthchild-X:Abelson Y:5 Z:116 Range: Abelson - Abelson
-lefthchild--rightchild-X:Anderson Y:0 Z:52 Range: Anderson - Anderson
-rightchild-X:Att-a Y:7 Z:71 Range: Aragon - Barto
-rightchild--lefthchild-X:Atre Y:0 Z:54 Range: Aragon - Atre
-rightchild--lefthchild--lefthchild-X:Aragon Y:0 Z:215 Range: Aragon - Aragon
-rightchild--rightchild-X:Barto Y:3 Z:105 Range: Barto - Barto

Delete node executed in: 1.2499999996862243e-05 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
4
Give the coordinates for the node to delete:
Enter X coordinate to delete: Tom
Enter Y coordinate to delete: 11
Enter Z coordinate to delete: 11
Tree after deleting the node:
X:Ladner Y:0 Z:251.0 Range: Aalst - Zdonik
-lefthchild-X:Fayed Y:0 Z:3.0 Range: Aalst - KÄ¶lling
-lefthchild--lefthchild-X:Cardelli Y:3 Z:250.0 Range: Aalst - Fahlman
-lefthchild--lefthchild--lefthchild-X:Abiteboul Y:0 Z:230.0 Range: Aalst - Cantrill
-lefthchild--lefthchild--lefthchild--lefthchild-X:Atre Y:0 Z:10.0 Range: Aalst - Bezos
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Allen Y:5 Z:37.0 Range: Aalst - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Abelson Y:6 Z:215.0 Range: Aalst - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Abebe Y:9 Z:41.0 Range: Aalst - Abiteboul
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Aaronson Y:7 Z:230.0 Range: Aalst - Aaronson
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild-X:Aalst Y:0 Z:1119.0 Range: Aalst - Aalst
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Abiteboul Y:2 Z:317.0 Range: Abelson - Abiteboul
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Abelson Y:5 Z:53.0 Range: Abelson - Abelson
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Ahm Y:2 Z:40.0 Range: Adleman - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Agrawal Y:9 Z:105.0 Range: Adleman - Agrawal
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild-X:Adleman Y:2 Z:54.0 Range: Adleman - Adleman
-lefthchild--lefthchild--lefthchild--lefthchild--lefthchild--rightchild--rightchild-X:Alfred Y:0 Z:113.0 Range: Alfred - Alfred
-lefthchild--lefthchild--lefthchild--lefthchild--rightchild-X:Aragon Y:0 Z:83.0 Range: Amdahl - Atanasoff
-lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild-X:Angie Y:0 Z:1.0 Range: Amdahl - Appel
-lefthchild--lefthchild--lefthchild--lefthchild--rightchild--lefthchild-X:Anderson Y:0 Z:573.0 Range: Amdahl - Anderson
```

```

-rightchild--rightchild--rightchild--rightchild--leftchild--leftchild-X:Worsley Y:5 Z:1.0 Range: Worsley - Worsley
-rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Wulf Y:0 Z:150.0 Range: Wulf - Wulf
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yung Y:12 Z:748.0 Range: Yao - Zdonik
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yoneda Y:0 Z:1.0 Range: Yao - Yourdon
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yen Y:0 Z:419.0 Range: Yao - Yao
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild-X:Yao Y:1 Z:179.0 Range: Yao - Yao
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild--leftchild-X:Younron Y:0 Z:22.0 Range: Youndon - Youndon
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Zantema Y:0 Z:118.0 Range: Zadeh - Zdonik
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Zaman Y:0 Z:35.0 Range: Zadeh - Zaman
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Zadeh Y:17 Z:237.0 Range: Zadeh - Zadeh
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zdonik Y:0 Z:194.0 Range: Zdonik - Zdonik

Delete node executed in: 5.4900000009183714e-05 seconds
Give a number for each choice below:

```

Αναζήτηση συγκεκριμένου κόμβου: Η συνάρτηση `searchNode()` παίρνει τη ρίζα του δέντρου εύρους 3D και τις συντεταγμένες (`x_coord`, `y_coord`, `z_coord`) του κόμβου που θέλω να αναζητήσω. Διασχίζει αναδρομικά το δέντρο με βάση τη συντεταγμένη `x`. Εάν η συντεταγμένη `x` του τρέχοντος κόμβου είναι μεγαλύτερη ή ίση με τη συντεταγμένη `x`-στόχο, πηγαίνει στο αριστερό υποδέντρο. Διαφορετικά, πηγαίνει στο δεξί υποδέντρο. Σε κάθε επίπεδο αναδρομής, ελέγχει εάν οι συντεταγμένες του τρέχοντος κόμβου ταιριάζουν με τις συντεταγμένες στόχου. Εάν το κάνουν, η συνάρτηση επιστρέφει `True`, υποδεικνύοντας ότι ο κόμβος με τις καθορισμένες συντεταγμένες βρίσκεται στο δέντρο. Εάν η συνθήκη συντεταγμένη `x` καθορίσει ότι η αναζήτηση πρέπει να συνεχιστεί στο αριστερό ή στο δεξί υποδέντρο, η συνάρτηση πραγματοποιεί μια αναδρομική κλήση με το κατάλληλο υποδέντρο. Εάν η ρίζα είναι `None` (που υποδεικνύει ένα κενό υποδέντρο), η συνάρτηση επιστρέφει `False`, καθώς ο κόμβος στόχος δεν μπορεί να βρεθεί σε ένα κενό δέντρο.

Στιγμιότυπο για το μικρό δέντρο:

```

5
Enter X coordinate to search: Atre
Enter Y coordinate to search: 0
Enter Z coordinate to search: 54
Node exists in the tree.

Search node executed in: 6.100000007336348e-06 seconds
Give a number for each choice below:

```

Στιγμιότυπο για το μεγάλο δέντρο:

```
5
Enter X coordinate to search: Fogel
Enter Y coordinate to search: 0
Enter Z coordinate to search: 25
Node exists in the tree.

Search node executed in: 1.2999999967178155e-05 seconds
```

Ενημέρωση κόμβου: Η συνάρτηση παίρνει τη ρίζα του δέντρου εύρους 3D και τις παλιές συντεταγμένες (old_x, old_y, old_z) του κόμβου που θέλω να ενημερώσω, καθώς και τις νέες συντεταγμένες (new_x, new_y, new_z). Η συνάρτηση λαμβάνει επίσης τρεις δυαδικές παραμέτρους (xtree, ytree, ztree) για να καθορίσει ποια διάσταση θα ληφθεί υπόψη κατά τη λειτουργία ενημέρωσης. Εάν το xtree είναι True, η ενημέρωση θα βασίζεται στη συντεταγμένη x και ομοίως για το ytree και το ztree. Η συνάρτηση διασχίζει αναδρομικά το δέντρο με βάση την καθορισμένη διάσταση (x, y ή z) μέχρι να βρει τον κόμβο με τις παλιές συντεταγμένες. Μόλις βρεθεί ο κόμβος με τις παλιές συντεταγμένες, η συνάρτηση ενημερώνει τις συντεταγμένες της στις νέες τιμές. Η συνάρτηση συνεχίζει να διασχίζει το δέντρο για να διατηρήσει τη δομή του. Εάν το xtree, το ytree ή το ztree είναι False, η συνάρτηση πραγματοποιεί επίσης μια αναδρομική κλήση για να ενημερώσει τη σχετική διάσταση στο δέντρο Y.

Στιγμιότυπο για το μικρό δέντρο:

```

6
Update a node:
Enter old X coordinate: Atta
Enter old Y coordinate: 7
Enter old Z coordinate: 71
Enter new X coordinate: Atta
Enter new Y coordinate: 8
Enter new Z coordinate: 81
Tree after updating the node:
X:Anthony Y:6 Z:317 Range: Abelson - Barto
-leftchild-X:Abramsky Y:6 Z:40 Range: Abelson - Anderson
-leftchild--leftchild-X:Abiteboul Y:3 Z:230 Range: Abelson - Abiteboul
-leftchild--leftchild--leftchild-X:Abelson Y:5 Z:116 Range: Abelson -
-leftchild--rightchild-X:Anderson Y:0 Z:52 Range: Anderson - Anderson
-rightchild-X:Atta Y:8 Z:81 Range: Aragon - Barto
-rightchild--leftchild-X:Atre Y:0 Z:54 Range: Aragon - Atre
-rightchild--leftchild--leftchild-X:Aragon Y:0 Z:215 Range: Aragon -
-rightchild--rightchild-X:Barto Y:3 Z:105 Range: Barto - Barto

Update node executed in: 1.0799999927257886e-05 seconds
Give a number for each choice below:

```

Στιγμιότυπο για το μεγάλο δέντρο:

```

9
Update a node:
Enter old X coordinate: tom
Enter old Y coordinate: 1
Enter old Z coordinate: 1
Enter new X coordinate: tom
Enter new Y coordinate: 11
Enter new Z coordinate: 11
Tree after updating the node:
X:Ladner Y:0 Z:251.0 Range: Aalst - tom
-leftchild-X:Fayed Y:0 Z:3.0 Range: Aalst - KÄ¶lling
-leftchild--leftchild-X:Cardelli Y:3 Z:250.0 Range: Aalst - Fahlman
-leftchild--leftchild--leftchild-X:Bhushan Y:0 Z:32.0 Range: Aalst - Cantrill
-leftchild--leftchild--leftchild-X:Atre Y:0 Z:0.0 Range: Aalst - Bezos
-leftchild--leftchild--leftchild--leftchild-X:Allen Y:5 Z:37.0 Range: Aalst - Atanasoff
-leftchild--leftchild--leftchild--leftchild--leftchild-X:Abromsky Y:6 Z:215.0 Range: Aalst - Alfred
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:abebe Y:9 Z:41.0 Range: Aalst - Abiteboul
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:aaronson Y:7 Z:230.0 Range: Aalst - Aaronson
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--leftchild-X:Aalst Y:0 Z:119.0 Range: Aalst - Aalst
-leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Abiteboul Y:2 Z:317.0 Range: Abelson - Abiteboul
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Abelson Y:5 Z:53.0 Range: Abelson - Abelson
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Ahn Y:2 Z:40.0 Range: Adleman - Alfred
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Agrawal Y:9 Z:105.0 Range: Adleman - Agrawal
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Adleman Y:2 Z:54.0 Range: Adleman - Adleman
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--rightchild-X:Alfred Y:0 Z:113.0 Range: Alfred - Alfred
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild-X:Atanasoff Y:0 Z:83.0 Range: Amdahl - Atanasoff
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Angie Y:0 Z:1.0 Range: Amdahl - Appel
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild-X:Anderson Y:0 Z:573.0 Range: Amdahl - Anderson
-leftchild--leftchild--leftchild--leftchild--leftchild--rightchild--leftchild--leftchild-X:Amdahl Y:4 Z:16.0 Range: Amdahl - Amdahl

```

```

-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild-X:Wolf Y:0 Z:150.0 Range: Wozniak - Yannakakis
-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--leftchild-X:Wu Y:0 Z:nan Range: Wozniak - Wu
-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--leftchild-leftchild-X:Wozniak Y:14 Z:13.0 Range: Wozniak - Wozniak
-rightchild--rightchild--rightchild--rightchild--leftchild--rightchild--rightchild-X:Yannakakis Y:1 Z:291.0 Range: Yannakakis - Yannakakis
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zadeh Y:17 Z:237.0 Range: Yen - tom
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Yourdon Y:0 Z:22.0 Range: Yen - Yung
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-Y:Yen Y:0 Z:1.0 Range: Yen - Yoneda
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-leftchild-X:Yung Y:12 Z:748.0 Range: Yung - Yung
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-X:Zdonik Y:0 Z:194.0 Range: Zaman - tom
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:Zantema Y:0 Z:118.0 Range: Zaman - Zantema
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild--leftchild-leftchild-X:Zaman Y:0 Z:35.0 Range: Zaman - Zaman
-rightchild--rightchild--rightchild--rightchild--rightchild--rightchild-X:tom Y:11 Z:11.0 Range: tom - tom

Update node executed in: 1.77000000078653e-05 seconds
Give a number for each choice below:

```

Αναζήτηση εύρους με Ish μέθοδο: Η συνάρτηση query_range() παίρνει τη ρίζα του δέντρου εύρους 3D και διάφορες παραμέτρους ερωτήματος όπως min_letter, max_letter, min_y, max_y, min_z και max_z. Ελέγχει εάν ο τρέχων κόμβος (ρίζα) ικανοποιεί τα κριτήρια του ερωτήματος με βάση τις συνθήκες που παρέχονται. Οι συνθήκες περιλαμβάνουν τον έλεγχο εάν η συντεταγμένη x εμπίπτει στο καθορισμένο εύρος γραμμάτων (min_γράμμα έως max_γράμμα), εάν η συντεταγμένη y είναι εντός του καθορισμένου εύρους (min_y έως max_y) και εάν η συντεταγμένη z είναι εντός του καθορισμένου εύρους (min_z έως max_z). Εάν πληρούνται οι προϋποθέσεις, η συνάρτηση ανακτά πληροφορίες εκπαίδευσης από το αρχικό σύνολο δεδομένων με βάση το επώνυμο (root.x) και προσαρτά ένα record που περιέχει τις συντεταγμένες και τις πληροφορίες εκπαίδευσης του κόμβου στη λίστα αποτελεσμάτων. Στη συνέχεια, η συνάρτηση καλείται αναδρομικά στο αριστερό και το δεξί υποδέντρο με βάση τη συντεταγμένη x και, εάν ισχύει, στο σχετικό δέντρο Y. Η αναδρομή συνεχίζεται έως ότου γίνει επίσκεψη όλων των κόμβων που πληρούν τα κριτήρια του ερωτήματος. Η λίστα αποτελεσμάτων τροποποιείται κατά την εκτέλεση της συνάρτησης και θα περιέχει πληροφορίες για όλους τους κόμβους που ικανοποιούν τις συνθήκες του ερωτήματος. Μετά την εκτέλεση της συνάρτησης εκτελούνται συγκεκριμένα βήματα που περιλαμβάνουν την εξαγωγή εκπαιδευτικών πληροφοριών από τα αποτελέσματα του ερωτήματος, την εφαρμογή Locality-Sensitive Hashing (LSH) για την εύρεση παρόμοιων εγγραφών με βάση τις πληροφορίες εκπαίδευσης τους και την εκτύπωση των παρόμοιων εγγραφών εκπαίδευσης.

Στιγμιότυπο για το μικρό δέντρο:

```

Enter the minimum letter limit from A to Z, a to z: 
Enter the maximum letter limit from A to Z, a to z: 
Enter the minimum awards value: 
Enter the maximum DBLPRecord value: 
Enter the minimum DBLPRecord value: 
Enter the maximum DBLPRecord value: 
Enter the minimum threshold (0 to 100): 
Similar Education Records:
("Abelson", 5, 11, "Abelson graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis, on Actions with fixed-point set: a homology sphere , supervised by William Browder .")
("Atta", 7, 71, nan)
("Abiteboul", 5, 230, "The son of two hardware store owners, Abiteboul attended high-school in Ramatamim , and Higher School Preparatory Classes in Tours . He was admitted to the Télécom Paris engineering school and studies at the Technion in Haifa for a year .")
("Barto", 3, 105, nan)
("Abramsky", 6, 40, "Abramsky was educated at Hasmonean Grammar School for Boys , Hendon and at King's College, Cambridge (BA 1975, MA Philosophy 1979, Diploma in Computer Science) and Queen Mary, University of London (PhD Computer Science 1988, supervised by Richard Bornat ).")
Create Range Tree executed in: 2.0000000233721948e-07 seconds

```

Στιγμιότυπο για το μεγάλο δέντρο:

```

/* Query search */
/*
Enter the minimum letter limit from A to Z, a to z:
Enter the maximum letter limit from A to Z, a to z:
Enter the minimum awards value:
Enter the minimum DBLPRecord value:100
Enter the maximum DBLPRecord value:1000
Enter the similarity threshold (0 to 100): 50
Similar Education Records:
('Pnueli', 6, 300.0, nan)
('Nivat', 3, 114.0, nan)
('Bachman', 5, 77.0, nan)
('Neumann', 3, 50.0, nan)
('Minker', 2, 157.0, nan)
('Borg', 3, 22.0, nan)
('Ho', 2, 112.0, 'Ho completed her undergraduate education at the Chinese University of Hong Kong in 1984. She received a Ph.D. in computer science from State University of New York at Buffalo in 1992.[1]')
('Kleinrock', 4, 188.0, nan)
('Perlmutter', 11, 68.0, 'As an undergraduate at MIT Perlman learned programming for a physics class. She was given her first paid job in 1971 as part-time programmer for the LOGO Lab at the (then) MIT Artificial Intelligence Laboratory, programming system software such as debuggers. [14]')
('Heiser', 14, 132.0, nan)
('Nagao', 10, 100.0, nan)
('Manz', 5, 192.0, nan)
('Edelman', 5, 90.0, 'Edelman received B.S. and M.S. degrees in mathematics from Yale University in 1984, and a Ph.D. in applied mathematics from MIT in 1989 under the direction of Lloyd N. Trefethen. Following a year at Thinking Machines Corporation, and at CERFACS[4] in France, Edelman went to U.C. Berkeley as a Morrey Assistant Professor and Levy Fellow, 1990a)x953. He joined the MIT faculty in applied mathematics in 1993.')
('Hennessy', 6, 122.0, nan)
('Barro', 2, 219.0, nan)
('Goldberg', 2, 163.0, nan)
('Hellman', 5, 76.0, nan)
('Kylling', 3, 46.0, nan)
*/

```

```

('Lampson', 19, 83.0, nan)
('Cardelli', 8, 250.0, 'He was born in Montecatini Terme, Italy. He attended the University of Pisa[7] before receiving his PhD from the University of Edinburgh in 1982[15] for research supervised by Gordon Plotkin.[4]')
('Feigenson', 10, 25.0, nan)
('Huang', 3, 28.0, nan)
('Glynnstra', 10, 124.0, nan)
('Kolter', 20, 266.0, 'Kolter received a bachelor's degree from the Hebrew University of Jerusalem in 1985, at the age of 17, and a master's degree from the same institution in 1986, at the age of 18.[11] She completed her PhD at Stanford in 1993 under the supervision of Joseph Halpern.[2]')
('Patt', 6, 219.0, nan)
('Pieraccini', 2, 85.0, 'He obtained a degree in electrical engineering from the University of Pisa in 1980 with a thesis on the equalization of data channels.')
('Create Range Tree executed in: 2.099998969246188e-07 seconds
Give a number for each choice below:

```

Πίνακας Σύγκρισης Αποτελεσμάτων

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
Range Δημιουργία δέντρου	0.00000040000000023354687	0.000000500000002290335
Range Αναζήτηση κόμβου	0.000006100000007336348	0.000012999999967178155
Range Εισαγωγή κόμβου	0.000009899999999340992	0.0000153999999522479
Range Ενημέρωση κόμβου	0.000010799999927257886	0.0000177000000078653
Range Διαγραφή κόμβου	0.000012499999996862243	0.00005490000009183714
Range Αναζήτηση εύρους (Ish)	0.0000020000000233721948	0.0000002999989692441886

QUAD-TREE: Το Quad-Tree είναι μια ιεραρχική δομή χωρικών δεδομένων. Αναπαριστά έναν δισδιάστατο χώρο χωρισμένο σε τεταρτημόρια. Κάθε κόμβος στο Quad-Tree αντιστοιχεί σε μια περιοχή του χώρου που ορίζεται από τα όριά του. Η ρίζα αντιπροσωπεύει ολόκληρο το χώρο και χωρίζεται σε τέσσερις κόμβους-παιδιά, καθένας από τους οποίους αντιπροσωπεύει ένα τεταρτημόριο.

Κλάση Node: Η κλάση Node αναπαριστά έναν μεμονωμένο κόμβο στο Quad-Tree. Κάθε κόμβος έχει τα ακόλουθα χαρακτηριστικά: value: Μια πλειάδα που αναπαριστά τα δεδομένα του επιστήμονα, η οποία περιέχει το επώνυμο του επιστήμονα και τον αριθμό των βραβείων. children: Μια λίστα με κόμβους-παιδιά. Στο Quad-Tree, κάθε κόμβος μπορεί να έχει έως και τέσσερα παιδιά. bounds: Μια πλειάδα που ορίζει τα χωρικά όρια της περιοχής του κόμβου.

Κλάση Quad-Tree: Η κλάση QuadTree αναπαριστά το ίδιο το Quad-Tree. Διαθέτει τα ακόλουθα χαρακτηριστικά: root: Η ρίζα του Quad-Tree, που αντιπροσωπεύει ολόκληρο το χώρο. bounds: Τα χωρικά όρια του Quad-Tree, που συνήθως ορίζονται ως πλειάδα με δύο σημεία (πάνω αριστερά και κάτω δεξιά). capacity: Ο μέγιστος αριθμός δεδομένων που μπορεί να χωρέσει ένας κόμβος πριν από τη διάσπαση σε κόμβους-παιδιά. (Η προεπιλεγμένη τιμή είναι 4.)

ΛΕΙΤΟΥΡΓΙΕΣ:

Δημιουργία δέντρου: Ο κώδικας ξεκινά με την αρχικοποίηση των ορίων του Quad-Tree. Αυτά τα όρια καθορίζουν τη χωρική έκταση του Quad-Tree. Σε αυτόν τον κώδικα, τα όρια ορίζονται ως μια πλειάδα με δύο σημεία, που συνήθως αντιπροσωπεύουν την πάνω αριστερή και την κάτω δεξιά γωνία ολόκληρου του χώρου. Για τα δεδομένα κάθε επιστήμονα (επώνυμο και βραβεία), ο κώδικας εισάγει έναν κόμβο στο Quad-Tree χρησιμοποιώντας τη μέθοδο insert(value, point) της κλάσης QuadTree. Η παράμετρος value για κάθε κόμβο περιέχει τα δεδομένα του επιστήμονα (επώνυμο, βραβεία και DBLP Record) και η παράμετρος point καθορίζεται με βάση τα βραβεία του επιστήμονα. Το Quad-Tree συμπληρώνεται με τα δεδομένα των επιστημόνων με αναδρομική διαίρεση του χώρου σε τεταρτημόρια και τοποθέτηση των δεδομένων κάθε επιστήμονα στον κατάλληλο κόμβο με βάση τα βραβεία του

```
C:\Users\sterg\IdeaProjects\untitled3\venv\Scripts\python.exe C:
Menu:
0) Exit the program
1) Make the quad-tree
2) Print the quad-tree
3) Insert a node in the quad-tree
4) Search the quad-tree for a certain node
5) Select a node to delete
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 1
Quad-tree created.

Creation of the tree executed in: 0.09748869994655252 seconds
```

Εκτύπωση δέντρου: Ο κώδικας εκτελεί τη μέθοδο print_tree(self, node, level) της κλάσης QuadTree. Αυτή η μέθοδος έχει σχεδιαστεί για να παρέχει μια οπτική αναπαράσταση της δομής του Quad-Tree. Λαμβάνει δύο παραμέτρους: node: Ο τρέχων κόμβος που υποβάλλεται σε επεξεργασία (ξεκινώντας από την ρίζα). level: Ένας ακέραιος αριθμός που δείχνει το βάθος ή το επίπεδο του τρέχοντος κόμβου στο Quad-Tree. Για κάθε κόμβο, εκτυπώνει την τιμή του κόμβου, η οποία περιέχει δεδομένα επιστήμονα (επώνυμο και βραβεία). Η συνάρτηση print_tree καλεί αναδρομικά τον εαυτό της στους κόμβους παιδιά του τρέχοντος κόμβου. Η αναδρομή συνεχίζεται μέχρι να εκτυπωθούν όλοι οι κόμβοι του Quad-Tree. Το αποτέλεσμα είναι μια οπτική αναπαράσταση της δομής του Quad-Tree, που δείχνει την ιεραρχία των κόμβων και τα χωρικά τους όρια

Στιγμιότυπο για το μικρό δέντρο:

```
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 2
('Atta', 10, 71)
('Aalst', 0, 1119)
('Aaronson', 7, 230)
('Abebe', 9, 0)
('Abelson', 5, 53)
('Abiteboul', 2, 317)
('Abramsky', 6, 215)
('Adleman', 2, 54)
('Agrawal', 9, 105)
('Ahn', 2, 40)
```

Στιγμιότυπο για το μεγάλο δέντρο:

/* Query search the quad-tree

```
Enter your choice: 2
('Atta', 10, '71')
('Alest', 0, '119')
('Aaronson', 7, '230')
('Abebe', 9, '41')
('Abelson', 5, '53')
('Abiteboul', 2, '317')
('Abramsky', 6, '215')
('Adleman', 2, '54')
('Agresti', 9, '105')
('Ahn', 2, '40')
('Alfred', 0, '113')
('Allan', 5, '37')
('Amahl', 4, '16')
('Anderson', 0, '573')
('Anthony', 0, '148')
('Appel', 0, '148')
('Aragon', 0, '83')
('Arden', 0, '37')
('Angle', 0, '1')
('Arons', 0, '245')
('Asprey', 0, '0')
('Atanasoff', 0, '46')
('Atre', 0, '0')
('Babbage', 0, '34')
('Bachman', 5, '77')
('Backhouse', 0, '71')
('Backus', 9, '25')
('Bacon', 0, '124')
('Bader', 0, '310')
('Bahl', 17, '129')
('Barn', 0, '18')
('Bartik', 4, '0')
('Barto', 2, '219')
('Bauer', 14, '196')
('Bayen', 0, '89')
('Bell', 0, '102')
('Bellovin', 0, '109')
```

('Berdichevsky', 0, '1')

```
('Berners-lee', 0, '80')
('Bennstein', 0, '259')
('Barus', 0, '73')
('Bezos', 0, '1')
('Bushan', 0, '32')
('Byj roer', 0, '100')
('Blasie', 0, '14')
('Black', 1, '1724')
('Biel', 4, '293')
('Blum', 0, '1')
('Blum', 0, '23')
('Blum', 0, '138')
('Boehm', 3, '396')
('Basm', 0, '33')
('Bollacker', 0, '24')
('Bonwick', 0, '7')
('Booch', 2, '132')
('Boote', 17, '53')
('Booth', 0, '50')
('Booth', 0, '3')
('Borg', 3, '22')
('Bos', 0, '197')
('Botvinnik', 0, '6')
('Boonen', 0, '308')
('Bourne', 0, '15')
('Bouwman', 0, '120')
('Boyer', 0, '78')
('Brandenburg', 1, '27')
('Bressard', 3, '134')
('Breed', 0, '11')
('Brenham', 0, '1')
('Brin', 0, '21')
('Brown', 0, '973')
('Brinch', 0, '59')
('Brinkemper', 0, '296')
('Brooks', 25, '180')
('Brooks', 0, '69')
('Burnett', 0, '264')
```

('Butter', 0, '200')

```
('Caballero Gil', 3, '214')
('Camp', 2, '129')
('Campbell-kelly', 0, '52')
('Candlin', 0, '4')
('Candan', 0, '0')
('Camtrill', 0, '13')
('Cardelli', 3, '250')
('Carneck', 0, '5')
('Casperson', 0, '62')
('Catmull', 0, '12')
('Catt', 38, '278')
('Chaitin', 0, '48')
('Callinan', 11, '13')
('Chauchan', 0, '41')
('Chen', 15, '3,209')
('Chiariglione', 3, '27')
('Chou', 0, '0')
('Church', 0, '21')
('Clararella', 0, '11')
('Clarke', 0, '524')
('Croke', 0, '485')
('Todd', 0, '56')
('Cohen', 0, '85')
('Coldwater', 0, '0')
('Colton', 0, '158')
('Colmeyer', 0, '33')
('Comer', 8, '91')
('Compton', 0, '0')
('Conaway', 0, '26')
('Gormack', 0, '133')
('Cook', 1, '234')
('Cooley', 0, '34')
('Cooper', 0, '2')
('CombatÃ©', 0, '17')
('Cosper', 0, '0')
('Cousot', 0, '125')
('Cox', 0, '179')
('Cray', 0, '2')
```

('Boller', 2, '7')
('Curry', 0, '17')
('Dadda', 0, '29')
('Dahl', 0, '42')
('Dan', 0, '27')
('Dan', 0, '101')
('Das', 0, '332')
('Dassani', 0, '15')
('Date', 0, '89')
('Davis', 0, '17')
('Dean', 6, '236')
('Demaine', 1, '742')
('DeMarco', 0, '43')
('Demillo', 0, '65')
('Denning', 9, '70')
('Denning', 0, '308')
('Dertouzos', 0, '17')
('Devaney', 0, '11')
('Dewan', 0, '54')
('Dewitt', 0, '10')
('Ditz', 0, '175')
('Dittrich', 4, '31')
('Dijkstra', 10, '124')
('Dillon', 0, '27')
('Disk', 0, '289')
('Dongarra', 5, '991')
('Dorigo', 0, '319')
('Dourish', 7, '100')
('Draper', 5, '0')
('Dumas', 1, '208')
('Dunkels', 0, '55')
('Dunn', 0, '13')
('Dusdorff', 0, '224')
('Eades', 0, '221')
('Eastey', 0, '0')
('Ebbinkhuizen', 0, '0')
('Eckert', 0, '9')
('Edelman', 5, '90')
('Eich', 0, '13')

('Barry', 0, '63')
('Baris', 0, '95')
('Bates', 0, '18')
('Beltzner', 0, '68')
('Gelobter', 0, '0')
('Besschke', 12, '12')
('Ohrmanni', 3, '322')
('Ghemawat', 6, '28')
('Gibbons', 0, '164')
('Gilbert', 0, '154')
('Giles', 2, '593')
('Ginsburg', 0, '126')
('Glass', 0, '371')
('Gädel', 3, '88')
('Goel', 0, '208')
('Gogen', 0, '172')
('Gold', 0, '283')
('Goldberg', 0, '61')
('Goldberg', 2, '163')
('Goldberg', 0, '155')
('Goldsmith', 0, '166')
('Goldreich', 1, '538')
('Goldwasser', 3, '239')
('Golub', 0, '170')
('Golumbic', 0, '110')
('Gonnet', 1, '102')
('Goodfellow', 0, '111')
('Gosling', 6, '16')
('Graham', 0, '468')
('Graham', 0, '278')
('Graham', 1, '92')
('Gray', 0, '270')
('Greifbach', 0, '77')
('Gries', 10, '164')
('Grilesem', 0, '101')
('Griswold', 0, '48')
('Gropp', 0, '310')
('Gruber', 0, '45')
('Gudeman', 0, '0')

('Heinzenberger', 0, '17')
('Weiser', 0, '76')
('Weizenauber', 0, '24')
('Wheeler', 0, '235')
('Westervelt', 0, '21')
('Wiltshire', 5, '208')
('Widom', 0, '23')
('Wiederhold', 0, '207')
('Wiener', 9, '34')
('Wijngaarden', 0, '49')
('Wilkes', 0, '2')
('Wilkes', 0, '84')
('Wilkes', 19, '231')
('Wilkinson', 5, '42')
('Wilson', 1, '18')
('Winograd', 0, '40')
('Winograd', 5, '122')
('Winston', 0, '61')
('Wirth', 0, '115')
('Wiseman', 0, '29')
('Wisniewsky', 0, '0')
('Wolfram', 0, '56')
('Woodger', 0, '0')
('Woodward', 0, '28')
('Worsley', 5, '13')
('Wozniak', 14, '13')
('Wu', 0, '7,065')
('Wolf', 0, '150')
('Yannakakis', 1, '291')
('Yao', 1, '179')
('Yen', 0, '419')
('Yoneida', 0, '11')
('Younion', 0, '22')
('Yung', 12, '768')
('Zaden', 17, '237')
('Zantema', 0, '210')
('Zaman', 0, '35')
('Zonnik', 0, '194')

Εισαγωγή νέου κόμβου στο δέντρο: Ο κώδικας δέχεται ως είσοδο το παρεχόμενο επώνυμο , τα βραβεία και το DBLP Record και δημιουργεί μια πλειάδα (surname,awards,DBLP_Record) που αντιπροσωπεύει τα δεδομένα του επιστήμονα. Στη συνέχεια καλείται η μέθοδος insert(value, point) της κλάσης QuadTree με αυτή την πλειάδα ως τιμή. Η παράμετρος point, η οποία καθορίζει πού τοποθετούνται τα δεδομένα του επιστήμονα στο Quad-Tree, υπολογίζεται με βάση την τιμή των βραβείων. Η μέθοδος insert της κλάσης QuadTree χειρίζεται τη διαδικασία εισαγωγής: Ξεκινώντας από την ρίζα, το Quad-Tree διαιρεί τον χώρο σε τεταρτημόρια αναδρομικά μέχρι να φτάσει σε έναν κατάλληλο κόμβο-φύλλο. Στη συνέχεια, τα δεδομένα του επιστήμονα (η πλειάδα (surname,awards,DBLP_Record) τοποθετούνται σε αυτόν τον κόμβο φύλλου.

Στιγμιότυπο για το μικρό δέντρο:

```
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 3
Enter the surname to insert: Stergiou
Enter the number of awards to insert: 10
Enter the DBLP record to insert: 10
Inserted (Stergiou, 10, 10) into the quad-tree.

Node inserted in: 2.3799948394298553e-05 seconds
('Atta', 10, 71)
('Aalst', 0, 1119)
('Aaronson', 7, 230)
('Abebe', 9, 0)
('Abelson', 5, 53)
('Abiteboul', 2, 317)
('Abramsky', 6, 215)
('Adleman', 2, 54)
('Agrawal', 9, 105)
('Ahn', 2, 40)
('Stergiou', 10, '10')
Menu:
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
('Wilkes', 0, '84')
('Wilks', 19, '211')
('Wilkinson', 5, '47')
('Wilson', 1, '18')
('Winograd', 0, '40')
('Winograd', 5, '122')
('Winston', 0, '61')
('Wirth', 0, '115')
('Wiseman', 0, '29')
('Wisniosky', 0, '0')
('Wolfram', 0, '56')
('Woodger', 0, '0')
('Woodward', 0, '26')
('Worsley', 5, '1')
('Wozniak', 14, '13')
('Wu', 0, '7,065')
('Wulf', 0, '150')
('Yannakakis', 1, '291')
('Yao', 1, '179')
('Yen', 0, '419')
('Yoneda', 0, '1')
('Youdon', 0, '22')
('Yung', 12, '748')
('Zadeh', 17, '237')
('Zantema', 0, '118')
('Zaman', 0, '35')
('Zdonik', 0, '194')
('Stergiou', 10, '10')
```

Αναζήτηση κόμβου στο δέντρο: Ο κώδικας δέχεται το επώνυμο, τα βραβεία και το DBLP Record ως δεδομένα εισόδου και ξεκινά μια αναζήτηση στο Quad Tree για έναν κόμβο με τα ίδια δεδομένα επιστήμονα. Καλεί τη μέθοδο search_node(node, surname, awards, dblp_record) της κλάσης QuadTree για να εκτελέσει την αναζήτηση. Η μέθοδος search_node είναι υπεύθυνη για τη διαδικασία αναζήτησης: Ξεκινώντας από την ρίζα του Quad-Tree, διατρέχει αναδρομικά το δέντρο, συγκρίνοντας τα δεδομένα του επιστήμονα σε κάθε κόμβο με το παρεχόμενο επώνυμο, βραβεία και DBLP record. Εάν βρεθεί ένας κόμβος με ταυτιαστά δεδομένα, η συνάρτηση επιστρέφει αυτόν τον κόμβο. Εάν κατά την αναζήτηση δεν βρεθεί κόμβος που να ταιριάζει, η συνάρτηση επιστρέφει None για να δηλώσει ότι ο επιστήμονας δεν υπάρχει στο QuadTree

Στιγμιότυπο για το μικρό δέντρο:

```
Menu:  
0) Exit the program  
1) Make the quad-tree  
2) Print the quad-tree  
3) Insert a node in the quad-tree  
4) Search the quad-tree for a certain node  
5) Select a node to delete  
6) Update a node with new values  
7) Query search the quad-tree  
Enter your choice: 4  
Enter the surname to search for: Stergiou  
Enter the number of awards to search for: 10  
Enter the DBLP record to search for: 10  
Node found: ('Stergiou', 10, '10')  
  
Found node in: 3.040023148059845e-05 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
Menu:  
0) Exit the program  
1) Make the quad-tree  
2) Print the quad-tree  
3) Insert a node in the quad-tree  
4) Search the quad-tree for a certain node  
5) Select a node to delete  
6) Update a node with new values  
7) Query search the quad-tree  
Enter your choice: 4  
Enter the surname to search for: Woodger  
Enter the number of awards to search for: 0  
Enter the DBLP record to search for: 0  
Node found: ('Woodger', 0, '0')  
  
Found node in: 0.0006830999627709389 seconds
```

Αν το όνομα δεν υπάρχει

```
ο) update a node with new values  
7) Query search the quad-tree  
Enter your choice: 4  
Enter the surname to search for: Abel  
Enter the number of awards to search for: 3  
Enter the DBLP record to search for: 3  
Node not found  
  
Found node in: 0.0006855996325612068 seconds
```

Διαγραφή επιλεγμένου κόμβου: Αφού λάβει το input του χρήστη, ο κώδικας ξεκινά τη διαδικασία διαγραφής καλώντας τη μέθοδο delete(surname, awards, dblp_record) της κλάσης QuadTree. Αυτή η μέθοδος είναι υπεύθυνη για τη διαγραφή των δεδομένων του επιστήμονα από το Quad-Tree. Η μέθοδος διαγραφής ξεκινά με την αναζήτηση στο Quad Tree για έναν κόμβο που ταιριάζει τόσο με το παρεχόμενο επώνυμο όσο και με τα βραβεία. Η διαδικασία αναζήτησης περιλαμβάνει τη διάσχιση του Quad-Tree από την ρίζα προς τα κάτω. Κατά τη διάρκεια αυτής της αναζήτησης, η μέθοδος συγκρίνει τα δεδομένα του επιστήμονα σε κάθε κόμβο (πλειάδα (surname, awards, dblp_record)) με τα παρεχόμενα κριτήρια για να εντοπίσει μια αντιστοιχία. Εάν βρεθεί ένας κόμβος που ταιριάζει με το παρεχόμενο επώνυμο και τα βραβεία, η μέθοδος delete προχωρά στην αφαίρεση του κόμβου από το Quad-Tree. Αυτή η ενέργεια διαγράφει ουσιαστικά τα δεδομένα του επιστήμονα από το σύνολο δεδομένων. Η διαδικασία διαγραφής περιλαμβάνει την αφαίρεση του κόμβου που ταιριάζει από τη λίστα των παιδιών του γονέα του. Εάν ο γονικός κόμβος δεν έχει άλλα παιδιά μετά τη διαγραφή, μπορεί επίσης να αφαιρεθεί, και αυτή η διαδικασία συνεχίζεται αναδρομικά προς τα πάνω στο δέντρο μέχρι να φτάσουμε σε έναν μη κενό γονέα ή στην ρίζα.

Στιγμιότυπο για το μικρό δέντρο:

```
Enter ,00, 0.010001 -  
('Atta', 10, '71')  
('Aalst', 0, '1,119')  
('Aaronson', 7, '230')  
('Abebe', 9, '41')  
('Abelson', 5, '53')  
('Abiteboul', 2, '317')  
('Abramsky', 6, '215')  
('Adleman', 2, '54')  
('Agrawal', 9, '105')  
('Ahn', 2, '40')  
Menu:  
0) Exit the program  
1) Make the quad-tree  
2) Print the quad-tree  
3) Insert a node in the quad-tree  
4) Search the quad-tree for a certain node  
5) Select a node to delete  
6) Update a node with new values  
7) Query search the quad-tree  
Enter your choice: 5  
Enter the surname to delete: Ahn  
Enter the number of awards to delete: 2  
Enter the DBLP record to delete: 40  
Deleted (Ahn, 2, 40) from the quad-tree.  
('Atta', 10, '71')  
('Aalst', 0, '1,119')  
('Aaronson', 7, '230')  
('Abebe', 9, '41')  
('Abelson', 5, '53')  
('Abiteboul', 2, '317')  
('Abramsky', 6, '215')  
('Adleman', 2, '54')  
('Agrawal', 9, '105')  
  
Node deleted in: 3.950018435716629e-05 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
0) Exit the program
1) Make the quad-tree
2) Print the quad-tree
3) Insert a node in the quad-tree
4) Search the quad-tree for a certain node
5) Select a node to delete
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 5
Enter the surname to delete: Wulf
Enter the number of awards to delete: 0
Enter the DBLP record to delete: 150
Deleted (Wulf, 0, 150) from the quad-tree.
('Atta', 10, '71')
('Aalst', 0, '1,119')
('Aaronson', 7, '230')
('Allison', 0, '111')
```

```
('Wijngaarden', 0, '49')
('Wilkes', 0, '2')
('Wilkes', 0, '84')
('Wilks', 19, '211')
('Wilkinson', 5, '47')
('Wilson', 1, '18')
('Winograd', 0, '40')
('Winograd', 5, '122')
('Winston', 0, '61')
('Wirth', 0, '115')
('Wiseman', 0, '29')
('Wisnosky', 0, '0')
('Wolfram', 0, '56')
('Woodger', 0, '0')
('Woodward', 0, '26')
('Worsley', 5, '1')
('Wozniak', 14, '13')
('Wu', 0, '7,065')
```

Όπως φαίνεται και στο στιγμιότυπο διαγράφεται ο συγκεκριμένος κόμβος μαζί με τα παιδιά του.

Ανανέωση επιλεγμένου κόμβου: Αφού λάβει το input του χρήστη, ο κώδικας ξεκινά τη διαδικασία ενημέρωσης καλώντας τη μέθοδο update_node(current_surname, current_awards, current dblp_record,new_surname, new_awards, new dblp_record) της κλάσης QuadTree. Η μέθοδος αυτή είναι υπεύθυνη για την τροποποίηση των δεδομένων του επιστήμονα που σχετίζονται με έναν συγκεκριμένο κόμβο στο Quad-Tree. Η μέθοδος update_node ξεκινά με την αναζήτηση στο Quad-Tree για έναν κόμβο που ταιριάζει τόσο με το παρεχόμενο current_surname όσο και με το current_awards και το current dblp_record. Η διαδικασία αναζήτησης περιλαμβάνει τη διάσχιση του Quad-Tree από την ρίζα προς τα κάτω. Κατά τη διάρκεια αυτής της αναζήτησης, η μέθοδος συγκρίνει τα δεδομένα του επιστήμονα σε κάθε κόμβο (πλειάδα (surname,awards,dblp_record)) με τα παρεχόμενα κριτήρια για να εντοπίσει μια αντιστοιχία. Εάν βρεθεί ένας κόμβος που ταιριάζει με τα παρεχόμενα current_surname , current_awards και current dblp_record, η μέθοδος update_node προχωρά στην τροποποίηση των δεδομένων που σχετίζονται με αυτόν τον κόμβο. Συγκεκριμένα, ενημερώνει τα δεδομένα του επιστήμονα, αλλάζοντας το τρέχον επώνυμο και τον τρέχοντα αριθμό βραβείων και DBLP Record στις νέες τιμές που παρέχονται από τον χρήστη.

Στιγμιότυπο για το μικρό δέντρο:

```
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 6
Enter the current surname of the node to update: Abramsky
Enter the current number of awards of the node to update: 6
Enter the current DBLP record of the node to update: 215
Enter the new surname: Stergiou
Enter the new number of awards: 7
Enter the new DBLP record: 7
Updated (Abramsky, 6, 215) to (Stergiou, 7, 7)
('Atta', 10, '71')
('Aalst', 0, '1,119')
('Aaronson', 7, '230')
('Abebe', 9, '41')
('Abelson', 5, '53')
('Abiteboul', 2, '317')
('Stergiou', 7, '7')
('Adleman', 2, '54')
('Agrawal', 9, '105')
('Ahn', 2, '40')

Node updated in: 3.079976886510849e-05 seconds
```

Στιγμιότυπο για το μεγάλο δέντρο:

```
4) Search the quad-tree for a certain node
5) Select a node to delete
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 6
Enter the current surname of the node to update: Wulf
Enter the current number of awards of the node to update: 0
Enter the current DBLP record of the node to update: 150
Enter the new surname: Stergiou
Enter the new number of awards: 7
Enter the new DBLP record: 7
Updated (Wulf, 0, 150) to (Stergiou, 7, 7)
('Atta', 10, '71')
('Aalst', 0, '1,119')
('Aaronson', 7, '230')
('Abebe', 9, '41')
('Abelson', 5, '53')
('Abiteboul', 2, '317')
('Abramsky', 6, '215')
('Adleman', 2, '54')
('Agrawal', 9, '105')
('Ahn', 2, '40')
('Alfred', 0, '113')
('Allen', 5, '37')
```

```
('Winograd', 0, '40')
('Winograd', 5, '122')
('Winston', 0, '61')
('Wirth', 0, '115')
('Wiseman', 0, '29')
('Wisniosky', 0, '0')
('Wolfram', 0, '56')
('Woodger', 0, '0')
('Woodward', 0, '26')
('Worsley', 5, '1')
('Wozniak', 14, '13')
('Wu', 0, '7,065')
('Stergiou', 7, '7')
('Yannakakis', 1, '291')
('Yao', 1, '179')
('Yen', 0, '419')
('Yoneda', 0, '1')
('Yourdon', 0, '22')
('Yung', 12, '748')
('Zadeh', 17, '237')
('Zantema', 0, '118')
('Zaman', 0, '35')
('Zdonik', 0, '194')
```

- Σε περίπτωση λανθασμένων δεδομένων από τον χρήστη

```
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 6
Enter the current surname of the node to update: giorgos
Enter the current number of awards of the node to update: 1
Enter the current DBLP record of the node to update: 1
Enter the new surname: giorgoss
Enter the new number of awards: 2
Enter the new DBLP record: 2
Node not found for updating

Node updated in: 0.0006985003128647804 seconds
```

Queries στο δέντρο: Όταν ο χρήστης επιλέξει την επιλογή "7", το πρόγραμμα ζητά από τον χρήστη να παράσχει τις ακόλουθες πληροφορίες: letter_range: (π.χ. [A, G]) που αντιπροσωπεύει το πρώτο γράμμα των επωνύμων που θα αναζητηθούν. Ο χρήστης εισάγει αυτό το εύρος ως συμβολοσειρά με τη μορφή "[A, G]". min_awards: Ο ελάχιστος αριθμός βραβείων που έλαβαν οι επιστήμονες για να συμπεριληφθούν στην αναζήτηση. Min dblp record: ο ελάχιστος αριθμός dblp record όπου έχει ένας επιστήμονας . Max dblp record: ο μέγιστος αριθμός dblp record που έχει ένας επιστήμονας. Η μέθοδος custom_search είναι υπεύθυνη για τη διαδικασία προσαρμοσμένης αναζήτησης: Ξεκινά φιλτράροντας τους επιστήμονες στο Quad-Tree με βάση τα καθορισμένα κριτήρια: Τα επώνυμα αρχίζουν με ένα γράμμα εντός της παρεχόμενης περιοχής letter_range. Επιστήμονες με βραβεία μεγαλύτερα ή ίσα με το καθορισμένο min_awards. Επιστήμονες με dblp record ανάμεσα στις τιμές των μεταβλητών min dblp record και max dblp record. Ο φιλτραρισμένος κατάλογος επιστημόνων αποθηκεύεται ως το αρχικό σύνολο αποτελεσμάτων. Μετά την εμφάνιση των αρχικών αποτελεσμάτων αναζήτησης, το πρόγραμμα προχωρά στην εκτέλεση αναζήτησης ομοιότητας βάσει της μεθόδου LSH στις παραγράφους εκπαίδευσης των επιστημόνων που υπάρχουν στα αρχικά αποτελέσματα. Η αναζήτηση LSH εντοπίζει επιστήμονες με παρόμοιες παραγράφους εκπαίδευσης με βάση την ομοιότητα συνημίτονου και σύμφωνα με το ποσοστό που εισαγάγει ο χρήστης. Εκτυπώνονται τα επώνυμα των επιστημόνων μαζί με τις παραγράφους εκπαίδευσής τους.

Στιγμιότυπο για το μικρό δέντρο:

```

5) Select a node to delete
6) Update a node with new values
7) Query search the quad-tree
Enter your choice: 7
Enter the letter range (e.g. [A,G]): [A,G]
Enter the minimum number of awards 1
Enter the minimum DBLP record 1
Enter the maximum DBLP record 600
Scientists with surnames in the range ['A', 'G'], more than 1 awards, and DBLP record between 1 and 600:
Enter the similarity percentage (0.0-100.0): 5
Scientists with similar education paragraphs:
Abelson Abelson graduated with a Bachelor of Arts degree in mathematics from Princeton University in 1969 after completing a senior thesis on Actions with fixed-point set: a homology sphere, supervised by William Browder. []
4
Abiteboul The son of two hardware store owners, Abiteboul attended high-school in Romorantin, and Higher School Preparatory Classes in Tours.[13] He was admitted to the Tâtelâcom Paris engineering school and studied at the 5
Abramsky Abramsky was educated at Hasmonean Grammar School for Boys, Hendon and at King's College, Cambridge (BA 1973, MA Philosophy 1977, Diploma in Computer Science) and Queen Mary, University of London (PhD Computer Sci
6
Query search executed in: 0.0019148801447319984 seconds

```

Στιγμιότυπο για το μεγάλο δέντρο:

```

Enter your choice: 7
Enter the letter range (e.g. [A,B]): [A,B]
Enter the minimum number of awards 1
Enter the minimum DBLP record 1
Enter the maximum DBLP record 600
Scientists with surnames in the range ['A', 'G'], more than 1 awards, and DBLP record between 1 and 600:
Enter the similarity percentage (0.0-100.0): 5
Scientists with similar education paragraphs:
Abiteboul The son of two hardware store owners, Abiteboul attended high-school in Romorantin, and Higher School Preparatory Classes in Tours.[13] He was admitted to the TÃ©lÃ©com Paris engineering school and studied at the
5
Abramsky Abramsky was educated at Hasnonean Grammar School for Boys, Hendon and at King's College, Cambridge (BA 1975, MA Philosophy 1979, Diploma in Computer Science) and Queen Mary, University of London (PhD Computer Scie
6
Anthony Lisa Anthony earned her B.S. and M.S. in computer science with official concentrations in artificial intelligence, human-computer Interaction, and software engineering at Drexel University. Her M.S. thesis involved
14
Booch Booch earned his bachelor's degree in 1977 from the United States Air Force Academy and a master's degree in electrical engineering in 1979 from the University of California, Santa Barbara.[8]
54
Brooks Born on April 19, 1931, in Durham, North Carolina,[7] he attended Duke University, graduating in 1953 with a Bachelor of Science degree in physics, and he received a Ph.D. in applied mathematics (computer science) fro
73
Cardelli He was born in Montecatini Terme, Italy. He attended the University of Pisa[7] before receiving his PhD from the University of Edinburgh in 1982[15] for research supervised by Gordon Plotkin.[4]
83
Cristianini Cristianini holds a degree in physics from the University of Trieste, a Master in computational intelligence from the University of London and a PhD from the University of Bristol.[2] Previously he has been a pr
115
Crowcroft Crowcroft was educated at Westminster School[11] and graduated with a Bachelor of Arts degree in physics in 1979 from the University of Cambridge where he was an undergraduate student of Trinity College, Cambridge
116
Croft Croft earned a bachelor's degree with honors in 1973 and a master's degree in computer science in 1974 from Monash University in Melbourne, Australia. He earned his Ph.D. in computer science from the University of Cra
117
Dean Dean received a B.S., summa cum laude, from the University of Minnesota in computer science and economics in 1990.[3] He received a Ph.D. in computer science from the University of Washington in 1996, working under
128
Finin Finin earned an undergraduate degree in Electrical Engineering from MIT in 1971 and a PhD in Computer Science from the University of Illinois at Urbana-Champaign in 1980.
171
Guha Guha did his schooling from Loyola High School, Pune and graduated with Bachelor of Technology in Mechanical Engineering from the Indian Institute of Technology Madras, Master of Science from University of California,
238

Query search executed in: 0.008442599792033434 seconds

```

Η παραπάνω λειτουργεία για να υλοποιηθεί χρησιμοποιεί μία ελάχιστα διαφορετική LSH μέθοδο από τις υπόλοιπες όπου:

LSH METHOD – QUAD-TREE

Η μέθοδος LSH δημιουργήθηκε ξεχωριστά για το συγκεκριμένο δέντρο του project. Παρακάτω θα εξηγηθεί με λεπτομέρεια ο τρόπος προσέγγισης, υλοποίησης και λειτουργίας της:

Δημιουργία συναρτήσεων κατακερματισμού:

Η συνάρτηση **create_hash_functions**(num_functions, num_buckets) δημιουργεί έναν κατάλογο συναρτήσεων κατακερματισμού. Η LSH χρησιμοποιεί πολλαπλές συναρτήσεις κατακερματισμού για να αντιστοιχίσει σημεία δεδομένων σε buckets με τρόπο ώστε παρόμοια σημεία δεδομένων να έχουν μεγαλύτερη πιθανότητα να αντιστοιχίσουν στο ίδιο bucket. Οι συναρτήσεις κατακερματισμού αναπαρίστανται ως πλειάδες (a, b), όπου a και b είναι τυχαία παραγόμενοι συντελεστές. Έπειτα

Η συνάρτηση **hash_value**(value, a, b, num_buckets) λαμβάνει μια τιμή (στην προκειμένη περίπτωση, την παράγραφο εκπαίδευσης ενός επιστήμονα), μαζί με τους συντελεστές της συνάρτησης κατακερματισμού a και b και τον αριθμό των buckets num_buckets. Εφαρμόζει τη συνάρτηση κατακερματισμού για τον υπολογισμό του κωδικού κατακερματισμού για τη δεδομένη τιμή και στη συνέχεια επιστρέφει τον δείκτη του bucket όπου πρέπει να τοποθετηθεί αυτή η τιμή.

Η συνάρτηση **create_lsh_buckets**(num_buckets) δημιουργεί ένα κενό λεξικό όπου τα κλειδιά αντιπροσωπεύουν δείκτες buckets και οι τιμές είναι αρχικά κενές λίστες. Αυτά τα buckets θα χρησιμοποιηθούν για την αποθήκευση των δεικτών των

επιστημόνων των οποίων οι παράγραφοι εκπαίδευσης έχουν κατακερματισμό στο ίδιο bucket.

Η συνάρτηση **cosine_similarity**(text1, text2) υπολογίζει την ομοιότητα συνημίτονου μεταξύ δύο text συμβολοσειρών. Χρησιμοποιεί την αναπαράσταση TF-IDF (Term Frequency-Inverse Document Frequency) του κειμένου για τον υπολογισμό της ομοιότητας. Μια υψηλότερη ομοιότητα συνημίτονου υποδηλώνει ότι οι δύο συμβολοσειρές κειμένου έχουν περισσότερες ομοιότητες.

Τέλος η συνάρτηση **lsh_education** είναι το κύριο μέρος της υλοποίησης της LSH. Ακολουθεί ο τρόπος λειτουργίας της: Αρχικά φilttrάρει τους επιστήμονες με βάση ένα καθορισμένο εύρος letter_range, min_awards, min_dblp_record και max_dblp_record. Δημιουργεί μια λίστα με συναρτήσεις κατακερματισμού και άδεια buckets. Επαναλαμβάνει την παράγραφο εκπαίδευσης κάθε επιστήμονα στη φilttrariσμένη λίστα και κατακερματίζει την παράγραφο χρησιμοποιώντας κάθε συνάρτηση κατακερματισμού. Οι προκύπτουσες τιμές κατακερματισμού καθορίζουν σε ποια buckets προστίθεται ο δείκτης του επιστήμονα. Μετά τον κατακερματισμό όλων των παραγράφων εκπαίδευσης, προσπαθεί να βρει παρόμοιες παραγράφους εκπαίδευσης με κατακερματισμό μιας προεπιλεγμένης παραγράφου εκπαίδευσης στόχου (στην προκειμένη περίπτωση, "Computer Science") χρησιμοποιώντας τις ίδιες συναρτήσεις κατακερματισμού. Όλοι οι επιστήμονες των οποίων οι παράγραφοι εκπαίδευσης κατατεμαχίζονται στους ίδιους κάδους με την παράγραφο-στόχο θεωρούνται πιθανές αντιστοιχίες. Στη συνέχεια, ο κώδικας υπολογίζει την ομοιότητα συνημίτονου μεταξύ των παραγράφων εκπαίδευσης ζευγαριών επιστημόνων των οποίων οι δείκτες συγκεντρώθηκαν στο σύνολο similar_scientists. Για κάθε ζεύγος επιστημόνων, υπολογίζει την ομοιότητα συνημίτονου μεταξύ των παραγράφων εκπαίδευσης χρησιμοποιώντας τη συνάρτηση cosine_similarity. Εάν η ομοιότητα είναι μεγαλύτερη ή ίση με το καθορισμένο κατώφλι similarity_threshold, προσθέτει τον δείκτη του πρώτου επιστήμονα του ζεύγους στον κατάλογο αποτελεσμάτων. Ο κώδικας χρησιμοποιεί έναν βρόχο για την επανάληψη της ταξινομημένης λίστας των δεικτών similar_scientists και συγκρίνει την παράγραφο εκπαίδευσης κάθε επιστήμονα με τον επόμενο επιστήμονα στην ταξινομημένη λίστα. Τέλος, η συνάρτηση επιστρέφει τη λίστα αποτελεσμάτων, η οποία περιέχει τους δείκτες των επιστημόνων των οποίων οι παράγραφοι εκπαίδευσης θεωρούνται παρόμοιες με βάση τα καθορισμένα κριτήρια.

Πίνακας Σύγκρισης Αποτελεσμάτων

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
QUAD Δημιουργία δέντρου	3.700004890561104e-05	0.09748869994655252
QUAD Αναζήτηση κόμβου	3.040023148059845e-05	0.000657899770885706
QUAD Εισαγωγή κόμβου	2.3799948394298553e-05	0.0004551997408270836
QUAD Ενημέρωση κόμβου	3.079976886510849e-05	0.0006585000082850456
QUAD Διαγραφή κόμβου	3.950018435716629e-05	0.0011208001524209976
QUAD Αναζήτηση εύρους (Ish)	0.00191480001447319984	0.008442599792033434

R-TREE

Η παρακάτω υλοποίηση αφορά την κατασκευή και επεξεργασία ενός R tree 3 διαστάσεων το οποίο περιέχει τα δεδομένα ενός πλήθους επιστημόνων, δηλαδή το όνομα , τον αριθμό των βραβείων τους και τον αριθμό των DBLP Record τους στην x, y και z διάσταση αντίστοιχα, τα οποία αντλούνται από ένα csv αρχείο.

Δημιουργούνται 2 κλάσεις , μια MinimumBoundingObject και μια RTree. Η 1η αναπαριστά ένα πλαίσιο (κουτί) , όπου και αποθηκεύονται τα δεδομένα του δέντρου, το οποίο εκτυπώνεται σε μορφή Minimum Bounding Objects. Το κάθε ένα από τα MBO περιέχει το πολύ 4 παιδιά και έχει 2 άκρα low και high. Η κλάση RTree επεξεργάζεται το δέντρο και όλες τις λειτουργίες του, όπως θα δούμε παρακάτω. Ο χρήστης καλείται να επιλέξει ανάμεσα σε 7 λειτουργίες όσον αφορά την επεξεργασία του δέντρου, οι οποίες παρουσιάζονται και αναλύονται παρακάτω. Να σημειωθεί πως η εκτέλεση και τα παραδείγματα γίνονται για 2 αρχεία με δεδομένα, ένα το οποίο περιέχει 10 επιστήμονες σύνολο και άλλο ένα με τους συνολικούς επιστήμονες. Σκοπός αυτού του διαχωρισμού των αρχείων είναι η ευκολότερη επεξεργασία και κατανόηση των λειτουργιών , ώστε να δουλεύουν τόσο σε μικρό όσο και σε μεγάλο όγκο δεδομένων.

ΛΕΙΤΟΥΡΓΙΕΣ

Δημιουργία δέντρου: Αρχικά, τα δεδομένα που θα αντλούνται από μια λίστα ταξινομούνται βάσει της 1ης στήλης του αρχείου, δηλαδή τα ονόματα των επιστημόνων, τα οποία αποθηκεύονται στην x συντεταγμένη, στην y αποθηκεύεται το πλήθος των βραβείων που έχει αποσπάσει ο καθένας και στην z το πλήθος των dblp όπου έχει ο καθένας. Χωρίζεται group κόμβων που έχουν ήδη ταξινομηθεί με μέγιστο πλήθος M, τα οποία μετά αποθηκεύονται σε ‘πλαίσια ’ minimum bounding objects. Τα ίδια τα minimum bounding objects προστίθενται σε μια λίστα

upper_level_items. Επειδή κάθε ένα MBO περιέχει το πολύ μέχρι M κόμβους, αν ένας κόμβος πρόκειται να προστεθεί σε ένα MBO, γίνεται διαχωρισμός και δημιουργείται ένα επιπλέον MBO. Κάθε MBO έχει lower και upper λίστες οι οποίες περιέχουν τις ακραίες τιμές του MBO. Αυτό υπολογίζεται στην συνάρτηση minimum_bounding_object_calculator, η οποία στο τέλος επιστρέφει το range του κάθε MBO στο δέντρο που υπολογίζεται ξεχωριστά όσον αφορά τις συντεταγμένες x, y και z. Να σημειωθεί ότι χρησιμοποιείται μια συνάρτηση update_rtree_and_list η οποία ενημερώνει κάθε φορά την λίστα από την οποία αντλούνται τα δεδομένα τα οποία αποθηκεύονται στο δέντρο κάθε φορά.

Στιγμιότυπο για το μικρό δέντρο:

```
// Query search
1
Root after tree creation: [<__main__.MinimumBoundingObject object at 0x000000
Create R Tree executed in: 9.699957445263863e-05 seconds
List Child: Low: ['Aalst', '0', '1,119'], High: ['Abiteboul', '9', '53']
    Leaf: ['Aalst', 0, '1,119']
    Leaf: ['Aaronson', 7, '230']
    Leaf: ['Abebe', 9, '41']
    Leaf: ['Abelson', 5, '53']
    Leaf: ['Abiteboul', 2, '317']
List Child: Low: ['Abramsky', '10', '105'], High: ['Atta', '9', '71']
    Leaf: ['Abramsky', 6, '215']
    Leaf: ['Adleman', 2, '54']
    Leaf: ['Agrawal', 9, '105']
    Leaf: ['Ahn', 2, '40']
    Leaf: ['Atta', 10, '71']
```

Εκτύπωση δέντρου: Στην εκτύπωση δέντρου εκτυπώνονται τα MBO και το range του κάθε MBO, καθώς και τα σημεία που περιέχει το καθένα. Γίνεται έλεγχος αν η ρίζα του δέντρου έχει παιδιά ή όχι (καθώς και γίνεται έλεγχος μέσω της isinstance αν το κάθε item ή root είναι MBO). Αν έχει, η συνάρτηση καλείται αναδρομικά για να εκτυπώσει τα παιδιά με indent+1 (indent ορίζει το βάθος του κάθε κόμβου και του κάθε MBO).

Στιγμιότυπο για το μικρό δέντρο:

```

7) Query search
2

Create R Tree executed in: 6.180023774504662e-05 seconds
List Child: Low: ['Aalst', '0', '1,119'], High: ['Abiteboul', '9', '53']
    Leaf: ['Aalst', 0, '1,119']
    Leaf: ['Aaronson', 7, '230']
    Leaf: ['Abebe', 9, '41']
    Leaf: ['Abelson', 5, '53']
    Leaf: ['Abiteboul', 2, '317']
List Child: Low: ['Abramsky', '10', '105'], High: ['Atta', '9', '71']
    Leaf: ['Abramsky', 6, '215']
    Leaf: ['Adleman', 2, '54']
    Leaf: ['Agrawal', 9, '105']
    Leaf: ['Ahn', 2, '40']
    Leaf: ['Atta', 10, '71']

```

Εισαγωγή κόμβου: Αρχικά γίνονται 2 έλεγχοι για τον κάθε κόμβο. Αν είναι αντικείμενο του MBO, τότε αποτελεί έναν κόμβο του δέντρου, κάτι το οποίο επαναλαμβάνεται για κάθε παιδί του κόμβου. Ύστερα γίνεται έλεγχος των z συντεταγμένων αν βρίσκονται ανάμεσα στις low και high τιμές. Αν ναι, τότε το new_children περιέχει το αποτέλεσμα την κλήσης της insert που γίνεται αναδρομικά με το παιδί item και τις νέες συντεταγμένες x, y και z. Αν υπάρχουν νέα παιδιά από την αναδρομική κλήση, τότε ενημερώνονται τα στοιχεία και τα όρια high,low του item. Σε αυτό το σημείο, εάν το πλήθος των παιδιών του κόμβου είναι μικρότερο από 4, δηλαδή το M, τότε προστίθεται στον κόμβο και ενημερώνονται τα όρια του. Αν όμως είναι μεγαλύτερο από 4, τότε το node κάνει split σε 2 MBO. Επιστρέφεται η λίστα με τα MBO και τα αντικείμενα μέσα σε αυτά. Στην άλλη περίπτωση που ο κόμβος δεν είναι MBO, τότε προστίθεται στην λίστα και μετατρέπεται στο τέλος σε MBO, ώστε στο τέλος να επιστραφούν όλα τα MBO με τα στοιχεία τους.

Στιγμιότυπο για το μικρό δέντρο:

```

6) Update a node
7) Query search
3
Give the coordinates to insert:
Enter Name: Babis
Enter Number of Awards: 10
Enter DBLP Record: 10
The updated R-tree after insertion:
List Child: Low: ['Aalst', '0', '1,119'], High: ['Abiteboul', '9', '53']
    Leaf: ['Aalst', 0, '1,119']
    Leaf: ['Aaronson', 7, '230']
    Leaf: ['Abebe', 9, '41']
    Leaf: ['Abelson', 5, '53']
    Leaf: ['Abiteboul', 2, '317']
List Child: Low: ['Abramsky', '10', '105'], High: ['Atta', '9', '71']
    Leaf: ['Abramsky', 6, '215']
    Leaf: ['Adleman', 2, '54']
    Leaf: ['Agrawal', 9, '105']
    Leaf: ['Ahn', 2, '40']
    Leaf: ['Atta', 10, '71']
List Child: Low: ['Babis', '10', '10'], High: ['Babis', '10', '10']
    Leaf: ['Babis', 10, 10]

Insert node executed in: 5.510030314326286e-05 seconds

```

Διαγραφή κόμβου: Αρχικά γίνεται έλεγχος αν ο κόμβος είναι αντικείμενο της κλάσης MBO. Δημιουργείται μια κενή λίστα new_children με τα παιδιά του κόμβου μετά την διαγραφή για κάθε παιδί του κόμβου node. Γίνεται αναδρομικός έλεγχος για το σημείο που θέλουμε να διαγράψουμε, και τα σημεία που δεν διαγράφονται προσθέτονται στην λίστα new_children. Αν το node_children είναι κενό, δηλαδή όλα τα παιδιά έχουν διαγραφεί, τότε διαγράφεται και ο κόμβος. Τέλος επιστρέφεται το νέο node.

Στιγμιότυπο για το μικρό δέντρο:

```

Leaf: ['Abramsky', 6, '215']
Leaf: ['Adleman', 2, '54']
Leaf: ['Agrawal', 9, '105']
Leaf: ['Ahn', 2, '40']
Leaf: ['Atta', 10, '71']
List Child: Low: ['Babis', '10', '10'], High: ['Babis', '10', '10']
Leaf: ['Babis', 10, 10]

Insert node executed in: 7.170019671320915e-05 seconds
Give a number for each choice below:
0) End the program
1) Create the R tree
2) Print the R tree
3) Insert a node
4) Delete a node
5) Search for a node
6) Update a node
7) Query search
4
Give the coordinates to delete:
Enter Name: Babis
Enter Number of Awards: 10
Enter DBLP Record: 10
The updated R-tree after deletion:
List Child: Low: ['Aalst', '0', '1,119'], High: ['Abiteboul', '9', '53']
    Leaf: ['Aalst', 0, '1,119']
    Leaf: ['Aaronson', 7, '230']
    Leaf: ['Abebe', 9, '41']
    Leaf: ['Abelson', 5, '53']
    Leaf: ['Abiteboul', 2, '317']
List Child: Low: ['Abramsky', '10', '105'], High: ['Atta', '9', '71']
    Leaf: ['Abramsky', 6, '215']
    Leaf: ['Adleman', 2, '54']
    Leaf: ['Agrawal', 9, '105']
    Leaf: ['Ahn', 2, '40']
    Leaf: ['Atta', 10, '71']

Delete node executed in: 8.999975398182869e-05 seconds

```

Αναζήτηση συγκεκριμένου κόμβου: Η αναζήτηση γίνεται αναδρομικά, καθώς εξετάζεται κάθε στοιχείο στον κόμβο node, και ελέγχεται αν είναι αντικείμενο της κλάσης MBO. Αν και οι 2 διαστάσεις του βρίσκονται στα όρια, τότε καλείται αναδρομικά η search_recursive για τα παιδιά, και μόλις κάποιο παιδί βρεθεί ίσο με το σημείο που ψάχνουμε, η μεταβλητή found γίνεται True.

Στιγμιότυπο για το μικρό δέντρο:

```
scientist's name: Maria
number of awards: 3
scientist's DBLP record: 3
ng node: (<__main___.MinimumBoundingObject object at 0x000001E70029F980>, <__main___.MinimumBoundingObject object at 0x000001E700126390>, <__main___.MinimumBoundingObject object at 0x000001E70029F680>], Target: ['Maria', 3,
:ng with child: <__main___.MinimumBoundingObject object at 0x000001E70029F980>
:ng with child: <__main___.MinimumBoundingObject object at 0x000001E700126390>
:ng with child: <__main___.MinimumBoundingObject object at 0x000001E70029F680>
```

Ενημέρωση κόμβου: Όσον αφορά την ενημέρωση ενός κόμβου, καλείται η delete για έναν ήδη υπάρχοντα κόμβο και ύστερα καλείται η αναδρομική insert για να γίνει εισαγωγή του νέου κόμβου στο σημείο του παλιού, καθώς και ενημερώνεται η λίστα με τα σημεία.

Στιγμιότυπο για το μικρό δέντρο:

```

Leaf: ['Atta', 10, '71']
List Child: Low: ['Babis ', '10', '10'], High: ['Babis ', '10', '10']
    Leaf: ['Babis ', 10, 10]

Insert node executed in: 7.440010085701942e-05 seconds
Give a number for each choice below:
0) End the program
1) Create the R tree
2) Print the R tree
3) Insert a node
4) Delete a node
5) Search for a node
6) Update a node
7) Query search
6
Enter the details of the scientist to update:
Enter scientist's current name: Babis
Enter scientist's current number of awards: 10
Enter scientist's current DBLP record: 10
Enter scientist's new name: Maria
Enter scientist's new number of awards: 3
Enter scientist's new DBLP record: 3
The updated R-tree after updating the node:
List Child: Low: ['Aalst', '0', '1,119'], High: ['Abiteboul', '9', '53']
    Leaf: ['Aalst', 0, '1,119']
    Leaf: ['Aaronson', 7, '230']
    Leaf: ['Abebe', 9, '41']
    Leaf: ['Abelson', 5, '53']
    Leaf: ['Abiteboul', 2, '317']
List Child: Low: ['Abramsky', '10', '105'], High: ['Atta', '9', '71']
    Leaf: ['Abramsky', 6, '215']
    Leaf: ['Adleman', 2, '54']
    Leaf: ['Agrawal', 9, '105']
    Leaf: ['Ahn', 2, '40']
    Leaf: ['Atta', 10, '71']
List Child: Low: ['Maria', '3', '3'], High: ['Maria', '3', '3']
    Leaf: ['Maria', 3, 3]

Update node executed in: 8.489983156323433e-05 seconds

```

LSH METHOD - RTREE

Στο συγκεκριμένο αλγόριθμο υλοποιήσαμε μία διαφορετική LSH συνάρτηση όπου χρησιμοποιεί

CLASS LSH

Η μέθοδος `__init__` αρχικοποιεί το αντικείμενο LSH με τον αριθμό των συναρτήσεων κατακερματισμού (num_hash_functions) και τον αριθμό των κάδων (num_buckets).

Καλεί επίσης τη μέθοδο `create_hash_functions` για τη δημιουργία των συναρτήσεων κατακερματισμού.

Create_hash_functions

Η μέθοδος `create_hash_functions` δημιουργεί τυχαίες συναρτήσεις κατακερματισμού (a και b) με βάση τον καθορισμένο αριθμό συναρτήσεων και κάδων.

Hash_value

Η μέθοδος `hash_value` υπολογίζει τους κατακερματιστές για μια δεδομένη τιμή (σε αυτή την περίπτωση, ένα διάνυσμα TF-IDF).

Επαναλαμβάνει για κάθε συνάρτηση κατακερματισμού και υπολογίζει τον κατακερματιστή χρησιμοποιώντας τον τύπο $(a * \text{hash}(\text{value}) + b) \% \text{num_buckets}$.

Query

Η μέθοδος `query` βρίσκει τα στοιχεία υποψήφιων στοιχείων στο ευρετήριο που είναι πιθανά παρόμοια με ένα δοσμένο διάνυσμα ερωτήματος και υπολογίζει τους κατακερματιστές για το διάνυσμα ερωτήματος χρησιμοποιώντας τη.

Έπειτα Η αναζήτηση εύρους γίνεται με τον εξής τρόπο: δέχεται αντικείμενα (`letter_range`, `min_awards`, `min dblp_record`, `max dblp_record`) για τις συντεταγμένες. Αποθηκεύονται στην `x`, `y` και `z` συντεταγμένη τα στοιχεία και ύστερα γίνεται έλεγχος εάν τα σημεία βρίσκονται εντός των ορίων. Αν βρίσκονται εντός των ορίων, τότε θα προστεθούν αυτά τα σημεία σε μία λίστα η οποία θα εκτυπωθεί στο τέλος. Η αναζήτηση εύρους δεν δουλεύει συνδυαστικά με την LSH λόγω κάποιας αδυναμίας του R-TREE , οπότε δεν παραθέτουμε κάποιο στιγμιότυπο.

Πίνακες Σύγκρισης Αποτελεσμάτων και Πειραμματικής Αξιολόγησης

KD

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
KD Δημιουργία δέντρου	0.000092000000000399	0.00956150000000223
KD Αναζήτηση κόμβου	0.0000342000000017606	0.0000417999996267725
KD Εισαγωγή κόμβου	0.00062700000000014875	0.006276899999889274

KD Ενημέρωση κόμβου	0.00022200000000793807	0.009578199999964454
KD Διαγραφή κόμβου	0.000027400000000454838	0.0000918999999157677
KD Αναζήτηση εύρους (Ish)	0.07121820000000056	0.0637988999999833

RANGE

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
Range Δημιουργία δέντρου	0.0000040000000023354687	0.00000500000002290335
Range Αναζήτηση κόμβου	0.00006100000007336348	0.0001299999967178155
Range Εισαγωγή κόμβου	0.0000989999999340992	0.00015399999522479
Range Ενημέρωση κόμβου	0.00010799999927257886	0.00017700000078653
Range Διαγραφή κόμβου	0.0001249999996862243	0.000549000009183714
Range Αναζήτηση εύρους (Ish)	0.000020000000233721948	0.000002999989692441886

QUAD

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)	Χρόνος εκτέλεσης για μεγάλο δέντρο (seconds)
QUAD Δημιουργία δέντρου	3.70004890561104e-05	0.09748869994655252
QUAD Αναζήτηση κόμβου	3.040023148059845e-05	0.000657899770885706
QUAD Εισαγωγή κόμβου	2.3799948394298553e-05	0.0004551997408270836
QUAD Ενημέρωση κόμβου	3.079976886510849e-05	0.0006585000082850456
QUAD Διαγραφή κόμβου	3.950018435716629e-05	0.0011208001524209976
QUAD Αναζήτηση εύρους (Ish)	0.00191480001447319984	0.008442599792033434

R

Μεθοδος	Χρόνος εκτέλεσης για μικρό δέντρο (seconds)
R-TREE Δημιουργία δέντρου	9.699957445263863e-05
R-TREE Αναζήτηση κόμβου	-
R-TREE Εισαγωγή κόμβου	5.510030314326286e-05
R-TREE Ενημέρωση κόμβου	8.489983156323433e-05
R-TREE Διαγραφή κόμβου	8.9999753981828669e-05
R-TREE Αναζήτηση εύρους (Ish)	-

ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρατηρώντας τους χρόνους που προκύπτουν από την πειραματική αξιολόγηση όπου φαίνεται και στα στιγμιότυπα , πως για μικρά δέντρα η δομή quad-tree είναι γρηγορότερη σε όλες τις μεθόδους εκτός της αναζήτησης εύρους μαζί με την Ish ενώ για μεγάλα δέντρα η δομή kd-tree είναι πιο γρήγορη σε σχέση με τις υπόλοιπες στη κατασκευή του δέντρου. Ωστόσο η δομή Range-Tree είναι πιο γρήγορη σε σχέση με τις υπόλοιπες σε όλες τις άλλες μεθόδους. Επίσης παρατηρούμε πως η πιο αργή υλοποίηση είναι αυτή του R-Tree η οποία είναι πιο αργή σε χρόνους ακόμα και από τις υλοποιήσεις για μεγάλα δέντρα των υπολοίπων δέντρων