2025

Πολυδιάστατες Δομές Δεδομένων

Project 1: Multi-dimensional Data Indexing and Similarity Query Processing

Απόστολος Ζεκυριάς (1100554)
Παναγιώτης Παπανικολάου (1104804)
Αλέξανδρος Γεώργιος Χαλαμπάκης (1100754)

Γενικές Πληροφορίες

Στις επόμενες σελίδες παρουσιάζονται οι απαντήσεις της ομάδας μας στο Project του μαθήματος "Πολυδιάστατες Δομές Δεδομένων". Σε αυτήν τη σελίδα παρέχονται πληροφορίες σχετικά με τα μέλη της ομάδας.

Η ομάδα αποτελείται από τους εξής φοιτητές:

Απόστολος Ζεκυριάς

Παναγιώτης Παπανικολάου

Αλέξανδρος Γεώργιος Χαλαμπάκης

Αναλυτικότερες Πληροφορίες:

Απόστολος Ζεκυριάς
1100554
up1100554@ac.upatras.gr
Φοιτητής 4ου έτους

Παναγιώτης Παπανικολάου
1104804
up1104804@ac.upatras.gr
Φοιτητής 4ου έτους

Αλέξανδρος Γεώργιος Χαλαμπάκης
1100754
up1100754@ac.upatras.gr
Φοιτητής 4ου έτους

1. Εισαγωγή

Σκοπός είναι η υλοποίηση και πειραματική αξιολόγηση τεσσάρων πολυδιάστατων δομών δεδομένων για το indexing και την αναζήτηση δεδομένων σε πολλαπλές διαστάσεις. Επιπλέον, υλοποιήθηκε η τεχνική LSH (Locality Sensitive Hashing) για την εύρεση ομοιότητας κειμένου στα αποτελέσματα των ερωτημάτων.

Για τις ανάγκες της εργασίας χρησιμοποιήθηκε το σύνολο δεδομένων "Movies Metadata Cleaned Dataset", το οποίο περιέχει πληροφορίες για ταινίες. Η υλοποίηση πραγματοποιήθηκε σε C.

2. Μεθοδολογία Υλοποίησης

2.1 Διαχείριση Δεδομένων (Data Loading)

Λόγω της πολυπλοκότητας του αρχείου CSV (ύπαρξη χαρακτήρων διαχωρισμού ; και χρήσης κόμματος , για δεκαδικά ψηφία), δημιουργήθηκε ένας CSV Parser σε C. Ο parser διαβάζει το αρχείο γραμμή προς γραμμή, αναγνωρίζει τις στήλες (Budget, Popularity, Runtime, Title, Genres) και μετατρέπει τα δεδομένα σε κατάλληλη μορφή (structs).

2.2 Δομές Δεδομένων

Υλοποιήθηκαν οι εξής δομές για το indexing 3 διαστάσεων (Budget, Popularity, Runtime):

1. k-d Tree**:** Δέντρο που διαχωρίζει τον χώρο εναλλάσσοντας τους άξονες σε κάθε επίπεδο.
2. Quad Tree**:** Προσαρμοσμένη υλοποίηση για τον τεμαχισμό του χώρου σε τεταρτημόρια, με φιλτράρισμα της 3ης διάστασης στα φύλλα (Υλοποιήθηκε ως **Octree** για την πλήρη κάλυψη των 3 διαστάσεων).
3. Range Tree**:** Δομή που ταξινομεί τα δεδομένα στην κύρια διάσταση και χρησιμοποιεί δευτερεύουσες δομές για τις υπόλοιπες (Με χρήση **Binary Search** στη δευτερεύουσα δομή για βελτιστοποίηση).
4. R-Tree**:** Δομή που ομαδοποιεί τα δεδομένα σε Ορθογώνια Ελάχιστου Περιβάλλοντος (MBRs) (Με μέθοδο **Bulk Loading** για ταχύτερη κατασκευή).

2.3 Αναζήτηση Ομοιότητας (LSH)

Για την εύρεση παρόμοιων ταινιών, υλοποιήθηκε η μέθοδος MinHash.

- Χαρακτηριστικό Κειμένου: Συνδυασμός Genres.

- Hash Functions: Χρησιμοποιήθηκαν 20 συναρτήσεις κατακερματισμού για τη δημιουργία της υπογραφής (signature) κάθε ταινίας.
- Ομοιότητα: Υπολογισμός Jaccard Similarity μεταξύ της ταινίας στόχου και των αποτελεσμάτων.

2.4 Πρόσθετες Λειτουργίες

Πέραν της βασικής αναζήτησης, υλοποιήθηκαν επιπλέον:

- k-Nearest Neighbors (kNN): Εύρεση των k πλησιέστερων γειτόνων βάσει Ευκλείδειας απόστασης.
- Delete Operation: Λειτουργία διαγραφής (Lazy Deletion) που αφαιρεί εγγραφές από τα αποτελέσματα χωρίς να απαιτεί πλήρη ανακατασκευή του δέντρου.

3. Πειραματική Αξιολόγηση

Τα πειράματα εκτελέστηκαν σε υπολογιστή με λειτουργικό σύστημα Windows, φορτώνοντας 200,000 εγγραφές ταινιών. Το ερώτημα εύρους (Range Query) που εκτελέστηκε ήταν κοινό για όλες τις δομές.

3.1 Πίνακας Αποτελεσμάτων

Ο παρακάτω πίνακας παρουσιάζει τους χρόνους κατασκευής (Build Time) και αναζήτησης (Query Time) για κάθε δομή.

Δομή Δεδομένων	Εγγραφές	Build Time (sec)	Query Time (sec)	Αποτελέσματα (Found)
k-d Tree	200,000	1.6800	0.0010	2798 → 2797
Quad Tree	200,000	0.1430	0.0010	2806 → 2805
Range Tree	200,000	0.7670	0.0040	2798 → 2797
R-Tree	200,000	0.1200	0.0020	2798 → 2797

Ανάλυση Κλιμακωσιμότητας (Scalability): Εκτελέστηκαν μετρήσεις αυξάνοντας το μέγεθος εισόδου (50k, 100k, 150k, 200k). Παρατηρήθηκε ότι ο χρόνος κατασκευής αυξάνεται γραμμικά, με το R-Tree να παραμένει η ταχύτερη δομή σε όλα τα μεγέθη.

3.2 Screenshots Εκτέλεσης

Ακολουθούν στιγμιότυπα από την εκτέλεση του κώδικα που επιβεβαιώνουν την ορθότητα των αποτελεσμάτων και τη λειτουργία του LSH.

1. Εκτέλεση k-d Tree & Αποτελέσματα

```
==================================
   C IMPLEMENTATION: MOVIE TREES
==================================
1. Run k-d Tree
2. Run Quad Tree
3. Run Range Tree
4. Run R-Tree
0. Exit
Choice: 1

--- Running k-d Tree ---
Loading data...
Loaded 200000 movies.

=== EXPERIMENTAL EVALUATION (Scalability) ===
| Dataset Size | Build Time (s) | Query Time (s) |
|--------------|----------------|----------------|
| 50000        | 0.1400         | 0.0010         |
| 100000       | 0.4350         | 0.0000         |
| 150000       | 0.9660         | 0.0000         |
| 200000       | 1.6800         | 0.0010         |

=== FULL DATASET OPERATIONS ===
Initial Query Results: 2798

[Delete Operation] Deleting 'The Toll Gate'...
Query Results after Delete: 2797 (Successfully removed)

[kNN Search] Top 5 Nearest Neighbors (Numeric Space) for 'Rouged Lips':
 2. Figures Don't Lie (Dist: 0.00)
 3. Queen of the Night Clubs (Dist: 0.01)
 4. Secret Service Investigator (Dist: 0.02)
 5. Mothers of Men (Dist: 0.02)
```

2. Εκτέλεση Quad Tree & LSH

```
================================
   C IMPLEMENTATION: MOVIE TREES
================================
1. Run k-d Tree
2. Run Quad Tree
3. Run Range Tree
4. Run R-Tree
0. Exit
Choice: 2

--- Running Quad Tree ---
Loading data...
Loaded 200000 movies.

=== EXPERIMENTAL EVALUATION (Scalability) ===
| Dataset Size | Build Time (s) | Query Time (s) |
|--------------|----------------|----------------|
| 50000        | 0.0340         | 0.0000         |
| 100000       | 0.0650         | 0.0010         |
| 150000       | 0.1050         | 0.0000         |
| 200000       | 0.1430         | 0.0010         |

[Delete Demo] Removing first result...
Count before: 2806, After: 2805

[kNN Search] Top 5 Nearest Neighbors (Numeric Space) for 'Mothers of Men':
 2. The Brass Bottle (Dist: 0.01)
 3. Queen of the Night Clubs (Dist: 0.01)
 4. Riding for Life (Dist: 0.02)
 5. Rouged Lips (Dist: 0.02)
```

3. Εκτέλεση Range Tree & R-Tree

```
==================================
   C IMPLEMENTATION: MOVIE TREES
==================================
1. Run k-d Tree
2. Run Quad Tree
3. Run Range Tree
4. Run R-Tree
0. Exit
Choice: 3

--- Running Range Tree ---
Loading data...
Loaded 200000 movies.

=== EXPERIMENTAL EVALUATION (Scalability) ===
| Dataset Size | Build Time (s) | Query Time (s) |
|--------------|----------------|----------------|
| 50000        | 0.1100         | 0.0020         |
| 100000       | 0.2660         | 0.0030         |
| 150000       | 0.4460         | 0.0030         |
| 200000       | 0.7670         | 0.0040         |

Found: 2798 movies (Should match 2798)

[Delete Demo] Removing first result: 'The hardest marriage'...
Count before: 2798, After: 2797

[kNN Search] Top 5 Nearest Neighbors (Numeric Space) for 'Mothers of Men':
 2. The Hope (Dist: 0.01)
 3. The Brass Bottle (Dist: 0.01)
 4. Queen of the Night Clubs (Dist: 0.01)
 5. Riding for Life (Dist: 0.02)

[LSH Similarity] Target: Mothers of Men
 -> Das Lied der Colombine (Sim: 1.00)
 -> The River's End (Sim: 0.45)
 -> Four Around the Woman (Sim: 1.00)
 -> The Old Nest (Sim: 1.00)
 -> Damon and Pythias (Sim: 1.00)
```

```
==================================
   C IMPLEMENTATION: MOVIE TREES
==================================
1. Run k-d Tree
2. Run Quad Tree
3. Run Range Tree
4. Run R-Tree
0. Exit
Choice: 4

--- Running R-Tree ---
Loading data...
Loaded 200000 movies.

=== EXPERIMENTAL EVALUATION (Scalability) ===
| Dataset Size | Build Time (s) | Query Time (s) |
|--------------|----------------|----------------|
| 50000        | 0.0150         | 0.0020         |
| 100000       | 0.0400         | 0.0020         |
| 150000       | 0.0720         | 0.0040         |
| 200000       | 0.1200         | 0.0020         |

[Delete Demo] Removing first result...
Count before: 2798, After: 2797

[kNN Search] Top 5 Nearest Neighbors (Numeric Space) for 'Das Lied der Colombine':
 2. Zoo in Budapest (Dist: 0.16)
 3. Sk?rg?rdsflirt (Dist: 0.16)
 4. Dreaming Lips (Dist: 0.38)
 5. Life Begins in College (Dist: 0.56)
```

4. Συμπεράσματα

Από την πειραματική διαδικασία προκύπτουν τα εξής συμπεράσματα:

1. Ορθότητα: Όλες οι δομές επέστρεψαν ακριβώς τον ίδιο αριθμό αποτελεσμάτων (2798), γεγονός που επιβεβαιώνει την ορθή υλοποίηση των αλγορίθμων αναζήτησης εύρους.
2. Χρόνος Κατασκευής: Το R-Tree και το Quad Tree αποδείχθηκαν τα πιο γρήγορα στη φάση κατασκευής (~0.1 sec)(σε κάθε εκτέλεση αλλάζουν λίγο οι χρόνοι), καθώς η

διαδικασία εισαγωγής ή bulk loading ήταν πιο άμεση. Το k-d Tree καθυστέρησε περισσότερο (1.68 sec) λόγω της ανάγκης ταξινόμησης (sorting/median finding) σε κάθε επίπεδο του δέντρου.

3. Χρόνος Αναζήτησης: Όλες οι δομές ανταποκρίθηκαν σχεδόν ακαριαία (< 0.01 sec), αποδεικνύοντας την αποτελεσματικότητά τους για μεγάλα σύνολα δεδομένων (200k εγγραφές).

4. Κλιμακωσιμότητα: Οι δομές ανταποκρίθηκαν αποδοτικά καθώς αυξανόταν ο όγκος δεδομένων, χωρίς εκθετική αύξηση του χρόνου.

Πρόσθετες Λειτουργίες: Η λειτουργία Delete και ο αλγόριθμος kNN επαληθεύτηκαν επιτυχώς, προσφέροντας δυνατότητα δυναμικής διαχείρισης και ευέλικτης αναζήτησης.

Η προσθήκη του LSH επέτρεψε τον εντοπισμό σημασιολογικά παρόμοιων ταινιών (π.χ. βάσει είδους) μέσα στο υποσύνολο των αποτελεσμάτων, προσφέροντας μια ολοκληρωμένη λύση αναζήτησης.