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.
3. Range Tree**:** Δομή που ταξινομεί τα δεδομένα στην κύρια διάσταση και χρησιμοποιεί δευτερεύουσες δομές για τις υπόλοιπες.
4. R-Tree**:** Δομή που ομαδοποιεί τα δεδομένα σε Ορθογώνια Ελάχιστου Περιβάλλοντος (MBRs).

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.

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	1.8550	0.0000	2798
Quad Tree	200,000	0.0810	0.0000	2798
Range Tree	200,000	0.8270	0.0060	2798
R-Tree	200,000	0.0770	0.0030	2798

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 from movies.csv...
Parsed [0]: Panorama of Galveston Power House | B:2427 P:6.37 R:1
Parsed [1]: Explosion of a Motor Car | B:1487 P:3.43 R:2
Parsed [2]: Panorama of Orphans' Home, Galveston | B:4687 P:3.24 R:1
Successfully loaded 200000 movies.
[k-d Tree] Building for 200000 movies...
Build Time: 1.8550 sec
[k-d Tree] Querying...
Query Time: 0.0000 sec | Found: 2798

[LSH Similarity] Target: The Toll Gate
```

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 from movies.csv...
Parsed [0]: Panorama of Galveston Power House | B:2427 P:6.37 R:1
Parsed [1]: Explosion of a Motor Car | B:1487 P:3.43 R:2
Parsed [2]: Panorama of Orphans' Home, Galveston | B:4687 P:3.24 R:1
Successfully loaded 200000 movies.
[Quad Tree] Building...
Build Time: 0.0810 sec
Query Time: 0.0000 sec | Found: 2798

[LSH Similarity] Target: Tenth Avenue Angel
 -> The Mystery of Edwin Drood (Sim: 0.50)
 -> King of Burlesque (Sim: 1.00)
 -> The Shanghai Story (Sim: 0.35)
 -> Wings of the Navy (Sim: 0.40)
```

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 from movies.csv...
Parsed [0]: Panorama of Galveston Power House | B:2427 P:6.37 R:1
Parsed [1]: Explosion of a Motor Car | B:1487 P:3.43 R:2
Parsed [2]: Panorama of Orphans' Home, Galveston | B:4687 P:3.24 R:1
Successfully loaded 200000 movies.
[Range Tree] Building...
Build Time: 0.8270 sec
Query Time: 0.0060 sec | Found: 2798

[LSH Similarity] Target: The hardest marriage
```

```
==================================
   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 from movies.csv...
Parsed [0]: Panorama of Galveston Power House | B:2427 P:6.37 R:1
Parsed [1]: Explosion of a Motor Car | B:1487 P:3.43 R:2
Parsed [2]: Panorama of Orphans' Home, Galveston | B:4687 P:3.24 R:1
Successfully loaded 200000 movies.
[R-Tree] Building...
Build Time: 0.0770 sec
Query Time: 0.0030 sec | Found: 2798

[LSH Similarity] Target: The hardest marriage
```

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, το Quad Tree και το Range Tree αποδείχθηκαν τα πιο γρήγορα στη φάση κατασκευής (~0.08 sec)(σε κάθε εκτέλεση αλλάζουν και οι χρόνοι και μπορεί να αλλάξουν θέσεις ως προς την ταχύτητα τους), καθώς η διαδικασία εισαγωγής ή bulk loading ήταν πιο άμεση. Το k-d Tree καθυστέρησε περισσότερο (1.85 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).

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