

Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών

HY240: Δομές Δεδομένων

Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2025-26

2η Φάση Προγραμματιστικής Εργασίας

Ημερομηνία Παράδοσης: 5 Δεκεμβρίου 2025, ώρα 23.59.

Τρόπος Παράδοσης: Μέσα από το eLearn του μαθήματος.

1. Γενική Περιγραφή

Στη δεύτερη φάση της εργασίας θα επεκτείνετε το σύστημα διαχείρισης βιβλιοθήκης που αναπτύχθηκε στην 1η φάση, ώστε να υποστηρίζει πιο σύνθετες λειτουργίες αναζήτησης, σύστασης και στατιστικής ανάλυσης.

Το νέο σύστημα θα βασίζεται σε **AVL**, **σωρούς (heaps)** και **δυναμικές λίστες δραστηριότητας**, προσφέροντας αποδοτικές αναζητήσεις και κατατάξεις.

Στόχοι της δεύτερης φάσης:

- Εμπέδωση σύνθετων δυναμικών δομών (δέντρα, σωροί).
- Διαχείριση δεικτών και δυναμικής μνήμης.
- Αποδοτικές λειτουργίες αναζήτησης και σύστασης σε $O(\log n)$.
- Ενοποίηση πληροφοριών από πολλαπλές δομές (συστάσεις, στατιστικά μελών, κατάταξη βιβλίων).

2. Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

Η δεύτερη φάση επεκτείνει το υπάρχον σύστημα, χωρίς να τροποποιεί τη βασική λειτουργικότητα της 1ης φάσης (δανεισμοί, επιστροφές, αξιολογήσεις, κατανομή θέσεων προβολής).

Προστίθενται **νέες δομές δεδομένων** και **νέες εντολές**, που βελτιώνουν την αποδοτικότητα και εμπλουτίζουν τις λειτουργίες.

3. Νέες Οντότητες και Δομές Δεδομένων

3.1. BookIndex (Δέντρο Αναζήτησης Βιβλίων)

Για την αποδοτική αναζήτηση βιβλίων κατά τίτλο, θα υλοποιήσετε ένα **ισορροπημένο AVL Tree**, στο οποίο κάθε κόμβος αντιστοιχεί σε ένα βιβλίο.

Πεδία (struct BookNode):

- title — Τίτλος βιβλίου (string).
- book — Δείκτης στη δομή Book του βιβλίου.

- `lc, rc` — Δείκτες σε αριστερό και δεξί παιδί.
- `height` — Ύψος κόμβου

Ιδιότητες:

- Το δέντρο είναι ταξινομημένο λεξικογραφικά κατά `title`.
- Επιτρέπεται μόνο ένα βιβλίο ανά τίτλο.
- Ενημερώνεται σε κάθε εισαγωγή/διαγραφή βιβλίου (BK ή R με `lost_flag=1`).

3.2. RecommendationHeap (Σωρός Συστάσεων)

Ο σωρός συστάσεων είναι ένας **Μέγιστος Σωρός (MaxHeap)** που περιέχει `capacity` δείκτες στα πιο δημοφιλή βιβλία όλων των κατηγοριών.

Πεδία (struct RecHeap):

- `heap[]` — Πίνακας δεικτών προς Book structs.
- `size, capacity`.

Κριτήριο προτεραιότητας: `avg` (μέση βαθμολογία).

Αν δύο βιβλία έχουν ίση βαθμολογία, υπερισχύει αυτό με μικρότερο `bid`.

Ο σωρός ενημερώνεται:

- Κατά την **επιστροφή βιβλίου (R)**, όταν αλλάζει η `avg`.
- Όταν βιβλίο δηλώνεται χαμένο, αφαιρείται από τον σωρό.

Ορίζουμε το `capacity` του RecommendationHeap σε 64, σταθερό κατά την διάρκεια εκτέλεσης του προγράμματος

Τα συνολικά βιβλία της βιβλιοθήκης επιτρέπεται να είναι παραπάνω από το `capacity` του RecommendationHeap

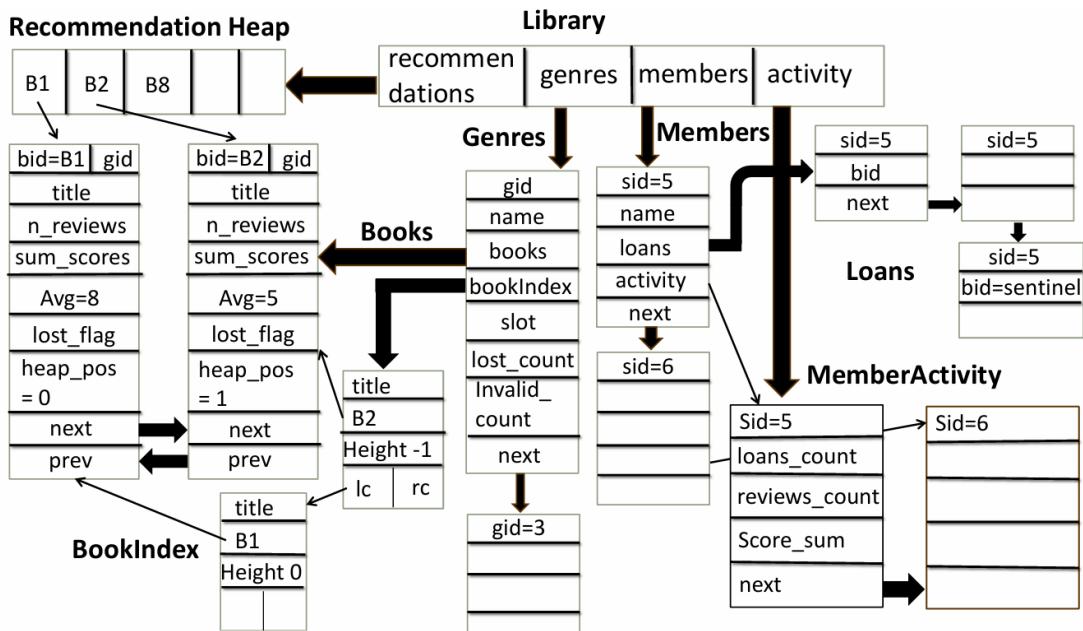
3.3. MemberActivity (Δραστηριότητα Μελών)

Κάθε μέλος παρακολουθείται ως προς τη δραστηριότητά του (πλήθος δανεισμών και αξιολογήσεων).

Πεδία (struct MemberActivity):

- `sid` — Κωδικός μέλους.
- `loans_count` — Πλήθος δανεισμών.
- `reviews_count` — Πλήθος αξιολογήσεων.
- `score_sum` — Σύνολο βαθμολογιών που έχει δώσει.
- `next` — Δείκτης στον επόμενο κόμβο λίστας.

Η λίστα είναι μη ταξινομημένη (unordered linked list) και ενημερώνεται με κάθε `L` και `R`.



4. Εντολές Προγράμματος

Το εκτελέσιμο παραμένει:

```
./library <input-file>
```

Κάθε γραμμή περιέχει μια εντολή. Οι νέες εντολές της 2ης φάσης είναι οι εξής:

| Εντολή | Περιγραφή | Ενέργεια / Έξοδος |
|---|--|--|
| F <title> | Αναζήτηση βιβλίου με βάση τον τίτλο στο BookIndex. | Εκτυπώνει Book <bid> "<title>" avg=<avg> ή NOT FOUND. |
| TOP <k> | Εμφάνιση των k κορυφαίων βιβλίων (παγκοσμίως) από τον RecommendationHeap. | Εμφανίζει Top Books: και λίστα <bid> "<title>" avg=<avg>. |
| AM | Εμφάνιση των πιο ενεργών μελών (με βάση πλήθος δανεισμών + αξιολογήσεων) χρησιμοποιώντας την MemberActivity list | Εκτυπώνει Active Members: και <sid> <name> loans=<L> reviews=<R>. |
| U <bid> "<new_title> " | Ενημέρωση τίτλου βιβλίου (επανατοποθέτηση στο BookIndex). | DONE ή IGNORED. |

| Εντολή | Περιγραφή | Ενέργεια / Έξοδος |
|--------|--------------------------------------|--|
| X | Εκτύπωση στατιστικών του συστήματος. | Stats: και τιμές για πλήθος βιβλίων, μελών, δανεισμών, μέσους όρους. |
| BF | Απελευθέρωση μνήμης όλων των δομών | DONE. |

5. Τροποποιήσεις Φάσης 1

5.1. Δομές Δεδομένων

- Στη δομή Library προστίθενται:
 - RecHeap* recommendations: δείκτης στον σωρό συστάσεων (μέγιστο σωρό).
 - MemberActivity* activity: δείκτης στη λίστα δραστηριότητας μελών.
- Στη δομή Book προστίθεται:
 - int heap_pos: θεση του book μεσα στο RecHeap. Αν το βιβλίο δεν ανήκει στο RecHeap τότε -1. (για αποδοτική ενημέρωση/ αφαίρεση του βιβλιου από το RecHeap).
- Στη δομή Member προστίθεται:
 - MemberActivity* activity: δείκτης στον αντίστοιχο κόμβο δραστηριότητας του μέλους (για γρήγορη ενημέρωση).

Οι υπόλοιπες δομές (Genre, Loan) παραμένουν ως έχουν.

5.2. Εντολές Προγράμματος

| Εντολή | Προσθήκη |
|---|---|
| BK <bid> <gid> <title> | Εκτός από την εισαγωγή στη λίστα της κατηγορίας, προσθέτει και στο BookIndex (AVL Tree) . |
| R <sid> <bid> <score> <status> | Εκτός από την επανατοποθέτηση στη λίστα της κατηγορίας: <ul style="list-style-type: none"> • Εάν αλλάζει η μέση βαθμολογία (score), ενημερώνει τον RecommendationHeap (αντί για απλή αναδιάταξη). • Εάν status=lost, αφαιρεί το βιβλίο από τον RecommendationHeap. • Ενημερώνει τη MemberActivity (αυξάνει το reviews_count αν δόθηκε score). |
| L <sid> <bid> | Εκτός από τη δημιουργία του δανεισμού, ενημερώνει τη MemberActivity (αυξάνει το loans_count) |
| M <sid> <name> | Εκτός από τη δημιουργία του μέλους, προσθέτει έναν νέο κόμβο στη MemberActivity list (αρχικοποιημένο). |

6. Απαιτήσεις Απόδοσης / Πολυπλοκότητας

| Εντολή / Λειτουργία | Απαιτούμενη Πολυπλοκότητα |
|---|---------------------------|
| Εισαγωγή/αναζήτηση στο BookIndex | $O(\log n)$ |
| Εισαγωγή/ενημέρωση στο RecommendationHeap | $O(\log n)$ |
| Ενημέρωση δραστηριότητας μέλους | $O(1)$ |
| Εκτύπωση TOP k | $O(k \log n)$ |
| Αναζήτηση F <title> | $O(\log n)$ |
| Καθαρισμός δομών (BF) | $O(n)$ |

7. Bonus

7.1. SystemStats

Διατήρηση μετρητών (πλήθος βιβλίων, μελών, ενεργών δανεισμάν, μέσος όρος όλων των αξιολογήσεων) που μπορούν να σας βοηθήσουν και στην διαχείριση των δομών δεδομένων.

| Εντολή | Περιγραφή | Ενέργεια / Έξοδος |
|--------|--------------------------------------|---|
| X | Εκτύπωση στατιστικών του συστήματος. | Stats: και τιμές για πλήθος βιβλίων, μελών, ενεργών δανεισμάν, και μέσο όρο αξιολογήσεων. |

7.2. MemberActivity Sorted Double-Linked list

Η λίστα MemberActivity μπορεί να υλοποιηθεί ως **ταξινομημένη διπλά συνδεδεμένη λίστα** (sorted double-linked list) με βάση το συνολικό σκορ δραστηριότητας (`loans_count + reviews_count`). Σε περίπτωση ισοβαθμίας, χρησιμοποιείται το `sid`.

- Κάθε κόμβος MemberActivity περιλαμβάνει τώρα:
 - `sid`, `loans_count`, `reviews_count`, `score_sum`
 - `next`: δείκτης στον επόμενο κόμβο (μικρότερο σκορ)
 - `prev`: δείκτης στον προηγούμενο κόμβο (μεγαλύτερο σκορ)
- Η εισαγωγή νέου μέλους γίνεται σε $O(n)$ (γραμμική εισαγωγή), αλλά η εντολή AM τυπώνει τα μέλη σε φθίνουσα σειρά δραστηριότητας σε $O(k)$
- Η ενημέρωση των `loans_count`, `reviews_count` οδηγεί σε `update` του κόμβου σε $O(hops)$ πολυπλοκότητα

8. Οδηγίες Παράδοσης

- Απαγορεύεται η χρήση εξωτερικών βιβλιοθηκών.
- Η εργασία παραδίδεται μέσω eLearn, με όλα τα απαραίτητα αρχεία πηγαίου κώδικα, του λάχιστον ένα αρχείο ελέγχου (`testfile`) που δείχνει τη σωστή

λειτουργία των νέων εντολών και ένα **Readme** ή **Makefile** με τον τρόπο που γίνεται compile η άσκηση στα **μηχανήματα της σχολής**