Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Δομές Δεδομένων
Εργασία 2

Διδάσκων: Δημήτρης Μιχαήλ

2020-2021

1 Εισαγωγή

Στην άσκηση αυτή καλείστε να υλοποιήσετε αλγορίθμους σύνδεσης (join) στην μνήμη. Είσοδος στο πρόγραμμα σας θα είναι τα αρχεία από την ιστοσελίδα `https://grouplens.org/datasets/movielens/1m/`. Το συγκεκριμένο dataset περιέχει τρία αρχεία (α) movies.dat, (β) ratings.dat και (γ) users.dat. Για τους σκοπούς της άσκησης θα χρειαστείτε μόνο τα αρχεία ratings.dat και movies.dat. Το αρχείο ratings.dat ακολουθεί το παρακάτω format:

`UserID::MovieID::Rating::Timestamp`

ενώ το αρχείο movies.dat το

`MovieID::Title::Genres`.

Σκοπός της εργασίας είναι να γράψετε ένα πρόγραμμα που να διαβάζει τα δύο αυτά αρχεία και να υπολογίζει στην έξοδο ένα αρχείο εξόδου της μορφής

`UserID::MovieID::Rating::Timestamp:Title::Genres`

το οποίο πρέπει να περιέχει μία γραμμή εξόδου για κάθε ζευγάρι γραμμών εισόδου (μία από κάθε αρχείο) που έχουν το ίδιο MovieID. Σε περίπτωση που ένα MovieID εμφανίζεται στο ένα μόνο αρχείο και όχι στο άλλο, δεν χρειάζεται να βγάλετε κάτι στην έξοδο για το συγκεκριμένο MovieID.

2 Παράδειγμα

Έστω πως το αρχείο εισόδου ratings.dat περιέχει τα εξής:

```
1::1::5::1611928841
1::2::3::1611928843
2::1::3::1611958848
3::1::2::1611948948
3::2::3::1611928747
3::3::2::1611938942
```

και πως το αρχείο movies.dat περιέχει τα εξής:

```
1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Ace Ventura: When Nature Calls (1995)::Comedy
```

Το αποτέλεσμα στο συγκεκριμένο παράδειγμα είναι:

```
1::1::5::1611928841::Toy Story (1995)::Animation|Children's|Comedy
1::2::3::16119288432::Jumanji (1995)::Adventure|Children's|Fantasy
2::1::3::1611958848::Toy Story (1995)::Animation|Children's|Comedy
3::1::2::1611948948::Toy Story (1995)::Animation|Children's|Comedy
3::2::3::16119287472::Jumanji (1995)::Adventure|Children's|Fantasy
3::3::2::1611938942::Ace Ventura: When Nature Calls (1995)::Comedy
```

3 Μέρος 1ο (6 βαθμοί)

Στο πρώτο μέρος της εργασίας καλείστε να υλοποιήσετε την παραπάνω διαδικασία χρησιμοποιώντας τον πιο απλό αλγόριθμο ωμής βίας (nested loop join), ο οποίος κάνει την εξής διαδικασία:

```
για κάθε γραμμή του αρχείου ratings.dat
  για κάθε γραμμή του αρχείου movies.dat
    αν το MovieID είναι ίδιο στις δύο γραμμές βγάλε αποτέλεσμα στην έξοδο
```

Αφού διαβάσετε μία γραμμή από τα αρχεία, μπορείτε να την σπάσετε σε κομμάτια χρησιμοποιώντας την `String.split()` ή την κλάση `StringTokenizer`.

Hint: Διαβάστε τα αρχεία πρώτα σε λίστες στην μνήμη και μετά εκτελέστε τον παραπάνω αλγόριθμο.

4 Bonus (1 βαθμός)

Χρησιμοποιήστε και το τρίτο αρχείο users.dat και βγάλτε ως αποτέλεσμα την σύνδεση (join) του αποτελέσματος των προηγούμενων ασκήσεων με το users.dat, αυτή την φορά χρησιμοποιώντας το UserID για την σύνδεση.

5 Μέρος 2ο (2.5 βαθμοί)

Δώστε μια υλοποίηση της παραπάνω διαδικασίας χρησιμοποιώντας πίνακα κατακερματισμού. Εδώ η τεχνική είναι η εξής: Χρησιμοποιείστε έναν πίνακα κατακερματισμού (HashMap) και εισάγετε όλες τις εγγραφές π.χ του movies. Χρησιμοποιήστε το MovieID ως κλειδί. Στην συνέχεια κάντε μία επανάληψη όλων των εγγραφών του ratings και για κάθε εγγραφή κάντε ερώτηση στον πίνακα κατακερματισμού για να δείτε αν υπάρχει αντίστοιχο MovieID. Αν ναι, βγάλτε στην έξοδο το αποτέλεσμα.

Hint: Χρειάζεται προσοχή με τα διπλότυπα, δηλαδή την περίπτωση όπου έχετε πολλές εγγραφές με το ίδιο κλειδί (MovieID). Αν προκύπτει, μπορείτε να χρησιμοποιήσετε συνδυασμό πίνακα κατακερματισμού και λιστών για να λύσετε αυτό το πρόβλημα.

6 Μέρος 3ο (1.5 βαθμοί)

Δώστε μία υλοποίηση της παραπάνω διαδικασίας με την χρήση της τεχνικής της συγχώνευσης (merge). Εδώ η τεχνική είναι να ταξινομήσετε στη μνήμη τα δεδομένα του κάθε αρχείου (με κλειδί το MovieID) και στην συνέχεια με ένα πέρασμα να βγάλετε την σωστή έξοδο ακολουθώντας την λογική της συγχώνευσης που είδαμε στο μάθημα. Για ταξινόμηση χρησιμοποιήστε τις έτοιμες συναρτήσεις που βρίσκονται στις κλάσεις Arrays και Collections.

7 Τεχνολογία

Για την υλοποίηση σας επιτρέπεται να χρησιμοποιήσετε μόνο την γλώσσα Java. Δεν επιτρέπεται όμως καμία χρήση βιβλιοθηκών εκτός από τις βασικές βιβλιοθήκες της γλώσσας. Επιτρέπεται να χρησιμοποιήσετε έτοιμες δομές όπως π.χ HashMap, HashSet, TreeMap, TreeSet, ArrayList, κ.τ.λ. καθώς και βοηθητικές συναρτήσεις π.χ για ταξινόμηση από τις κλάσεις Arrays και Collections καθώς και τις βοηθητικές συναρτήσεις στην κλάση Comparator.

8 Παράδοση

Το παραδοτέο της άσκησης είναι μόνο ένα αρχείο πηγαίου κώδικα σε γλώσσα Java. Αυτό βέβαια δεν σημαίνει πως δεν έχετε υποχρεώση να γράψετε ωραίο κώδικα ο οποίος χρησιμοποιεί σωστά συναρτήσεις και κλάσεις. Για να χρησιμοποιήσετε επιπλέον κλάσεις μπορείτε να τις φτιάξετε ως inner classes. Αν θέλετε υποχρεωτικά να χρησιμοποιήσετε πολλά αρχεία πηγαίου κώδικα, ο μόνος τρόπος για να γίνει αποδεκτή η άσκηση σας είναι να χρησιμοποιήσετε Maven project όπως κάναμε στο εργαστήριο. **Προσοχή, η βαθμολόγηση δεν γίνεται μόνο με βάση την λειτουργικότητα αλλά και με βάση την ποιότητα του κώδικα.**

9 Λογοκλοπή

Η άσκηση αυτή είνα αυστηρά προσωπική. Οποιαδήποτε μορφή λογοκλοπής από το internet ή από συνάδελφο σας θα μηδενίζεται απευθείας. Επίσης προσοχή πως σε περίπτωση αντιγραφής θα μηδενίζονται απευθείας **όλοι** οι εμπλεκόμενοι.

Καλή επιτυχία!