Διαχείριση Σύνθετων Δεδομένων

1ο Σετ ασκήσεων Αναφορά

Μπόζ Ντουράν

A.M 2310

Γλώσσα υλοποίησής : Python

Βιβλιοθήκες που χρησιμοποιήθηκαν : timeit ,operator

Για το κάθε ερώτημα υπάρχει και ένα διαφορετικό αρχείο με το αντίστοιχο όνομα.

Τα 3 πρώτα ερωτήματα τα αποτελέσματα αποθηκεύονται σε αρχεία txt.

Τα υπόλοιπα 2 τυπώνονται στο τερματικό.

Q1.1 -) Στη πρώτη ερώτηση χρησιμοποιώ merge join απαραίτητη προϋπόθεση είναι το αρχείο να είναι ταξινομημένο ως προς τα κοινά στοιχεία .Διαβάζω τα αρχεία γραμμή για να μην χρησιμοποιούμαι πολύ μνήμη και λόγο του μεγέθους των αρχείων η φόρτωση ολοκλήρου του αρχείο μπορεί να μην είναι δυνατή. Διαβάζω όλες της γραμμές μέχρι το τέλος και αυτές που ικανοποιούν την ερώτηση της γραφώ μια μια στο αρχείο txt.

- Input files: title.basics.tsv , title.crew.tsv
- Output File: outputQ1_1.txt
- Άλγεβρα: **π** primaryTitle,directors(title.basics **|><|** titlecrew)
  - tconst=tconst **and** director >2

Q1.2-)Χρήση merge join σε ταξινομημένα αρχεία . Διαβάζω γραμμή γραμμή τα αρχεία μέχρι το τέλος και των 2.Οι γραμμές που ικανοποιούν την ερώτηση γράφονται στο αρχείο txt.

- Input files: title.basic.tsv , title.episode.tsv
- Output File: outputQ1_2.txt
- Άλγεβρα**: π** primaryTitle,parenTconst,seasonNumber (title.basic **|><|** title.episode)

  tconst=tconst **and** episodeNumber=1

Q1.3-) Χρήση merge join σε ταξινομημένα αρχεία . Διαβάζω γραμμή γραμμή τα αρχεία μέχρι το τέλος και ων 2.Οι γραμμές που ικανοποιούν την ερώτηση γράφονται στο αρχείο txt.

- Input files: title.basics.tsv ,title.ratings.tsv
- Output File: outputQ1_3.txt
- Άλγεβρα: **π** primaryTitle(title.basics **|><|** title.ratings)
  tconst=tconst **and** averageRating = \N

Q2.1-)

- Q2.1 .sorted -)Διαβάζω όλο το αρχείο και το αποθηκεύω στη μνήμη και το κάνω sort.Έπειτα συγκρίνω και αποθηκεύω τα αποτελέσματα σε μια λίστα που την χρησιμοποιώ ως counter.Στο τέλος κάνω print τα αποτελέσματα.
- Q2.1.hased -)Διαβάζω το αρχείο γραμμή γραμμή δεν χρειάζεται να αποθηκεύσω το αρχείο στη μνήμη καθώς μπορώ να έχω τα αποτελέσματα κατευθείαν χρησιμοποιώντας μια hash function και ένα dictionary ως hash table.Στο hash table αποθηκεύω σαν value τον πλήθος των ratings που ικανοποιούν της συνθήκες και key το αποτέλεσμα της hash function.Στο τέλος κάνω print τα αποτελέσματα.
- Για την χρονομέτρηση χρησιμοποίησα τη βιβλιοθήκη timeit.Η υλοποίηση της Q21hash κάνει σχεδόν το μισό χρόνο ,και γλιτώνουμε την φόρτωση του αρχείου στη μνήμη .Η διαφορά θεωρό είναι ότι γλιτώνουμε το να κάνουμε sort το αρχείο και την μνήμη που θα χριζόμασταν για να φορτώσουμε όλο το αρχείο.
- Input file : title.ratings.tsv
- Άλγεβρα :

Q2.2-)Για την υλοποίηση της Q2.2 χρησιμοποίησα merge join.Σε ένα dictionary αποθήκευα τα ratings τον ταινιών και τον πλήθος τον ταινιών για κάθε χρονιά Στο τέλος διαιρούσα το συνολικό rating με τον πλήθος τον ταινιών .Τα αρχεία τα διαβάζω γραμμή γραμμή. Στο τέλος κάνω print το αποτέλεσμα.

- Input file: title.basics.tsv , title.ratings.tsv
- Αλγεβρα R <- **σ** (title.basics **|><|** title.ratings)
  tconst=tconst

  R1<- startYear **g** sum(averageRating) (R)

  R2<- startYear **g** count(startYear) (R)

Τα στοιχεία του R1 και του R2 χρειάζονται διαίρεση ώστε να παράγουν το τελικό αποτέλεσμα.