Movie.java

Movie(int, int, String): constructor ο οποίος δίνει τιμές στο id, likes, title.

getId(), getLikes(),getTitle(): επιστρέφουν τις τιμές των id, likes, title.

compareTo(Movie): συγκρίνει το τρέχον αντικείμενο(this) με το ορισμα

με βάση τα likes , αν τα likes τους είναι ίδια τότε τα συγκρίνει ανάλογα με την αλφαβητική ταξινόμηση των title. Αν είναι το τρέχων αντικείμενο μεγαλύτερο τοτε επιστρέφει 1, αλλιώς -1

Top_k.java

readFile(): Διαβάζει το αρχείο πρώτα γραμμή- γραμμή για να βρεί τον αριθμό των ταινιών , και να δημιουργήσει πίνακα με το κατάληλο μέγεθος. Έπειτα διαβάζει το αρχείο χαρακτήρα χαρακτήρα για να ξεχωρίσει τα id, likes και title της καθε ταινίας. Αφού δημιουργήσει το αντίστοιχο αντικείμενο τύπου Movie το καταχωρεί στον πίνακα. Τέλος μόλις γεμίσει τον πίνακα και τελιώσει το διάβασμα του αρχείου, καλεί την *quickSort()* τον ταξινομήσει.

*quickSort():*Η quickSort είναι μια μέθοδος ταξινόμησης η οποία παίρνει για όρισμα 2 ακεραίους αριθμούς ,στην πράξη η quicksort χωρίζει έναν πίνακα σε δυο μέρη με μια μεταβλητή pivot η οποία στην προκείμενη περίπτωση βρίσκεται πάντα στην μέση του πίνακα -υποπινακα που ταξινομείται , στην ουσία η quicksort καλείτε αναδρομικά έως ότου τα στοιχεία αριστερά από το pivot να είναι μικρότερα του και δεξιά του pivot μεγαλύτερα του

PQ.java

Size(): Επιστρέφει το σύνολο των “ενεργών” στοιχείων του πίνακα

isEmpty(): Επιστρέφει την τιμή true όταν ο πίνακας είναι άδειος και την τιμή false όταν περιέχει στοιχεία

insert(): Εισάγει ένα νέο αντικείμενο στον πίνακα και ταξινομεί την ουρά προτεραιότητας, μαζί με το νέο αντικείμενο.

getMax(): Επιστρέφει και αφαιρεί το στοιχείο που βρίσκεται στην θέση 1 του πίνακα

swim(): H swim χρησιμοποιείται αφού εισαχθεί ένα νέο στοιχείο προκειμένου να αναδιοργανώσει τον πίνακα στην κατάλληλη κατάταξή (φθίνουσα ,αύξουσα)

sink(): H sink χρησιμοποιείται αφού αφαιρεθεί κάποιο στοιχείο προκειμένου να αναδιοργανώσει τον πίνακα στην κατάλληλή κατάταξη(φθίνουσα ,αύξουσα)

resize(): Διπλασιάζει το μέγεθος του πίνακα όταν τα ενεργά στοιχεία του φτάσουν στο 75% της συνολικής χωρητικότητας

swap(): Αντιστρέφει 2 αριθμούς

Max(): Επιστρέφει τα στοιχείο που βρίσκεται στην θέση 1 του πίνακα χωρίς όμως να το αφαιρει

findMax(): Αυτή η μέθοδος καλείται από την sink προκειμένου να επιστρέψει το μεγαλύτερο από τα δυο ορίσματα που δέχεται

Top_k_withPQ.java

Διαβάζει απο τον χρήστη τον αριθμό των ταινιών που θέλει να εμφανίσει και δημιουργεί ουρά προτεραιότητας

readFile(): Όμοια με το Top_k διαβάζει το αρχείο και ξεχωρίζει τα id, likes και title κάθε ταινίας. Μόλις φτιάξει το αντικείμενο και η ουρά δεν ειναι γεμάτη το εισάγει, αλλιώς συγκρίνει το αντικείμενο με το μικρότερο της ουράς(Max) το οποίο θα έχει τα λιγότερα likes και αν έχει περισότερα, το εισάγει στην ουρά και αφαιρεί το τρέχον max.

Dynamic_Median.java

Όπως και οι παραπάνω κλάσεις στην readFile() διαβάζει το αρχείο και σε κάθε επανάληψη δημιουργεί αντικείμενο Movie. Στην πρώτη επανάληψη το εισάγει στην ουρά που θα έχει το μεγαλύτερο μισό των likes(bigger). Στην 2η το βάζει στην άλλη ουρα(lesser). Στις επόμενες επαναλήψεις συγκρίνει τις ταινίες που είναι στα Max της κάθε ουράς και το τοποθετεί ανάλογα(αν είναι μικρότερο απο το max της lesser, το τοποθετεί στην lesser, αλλιώς στην bigger, ελέγχοντας ομως αν η bigger ξεπερνά σε μέγεθος την lesser πανω απο 1. Οι δύο ουρές πρέπει να είναι παρόμοιες σε μέγεθος, δηλ. Ή ίσες , ή η bigger να είναι κατα ένα μεγαλύτερη ετσι ώστε το max της bigger να δείχνει το τρέχων median.). Κάθε 5 επαναλήψεις εμφανίζει τον τρέχων median και την ταινία η οποία αντιστοιχεί στα likes.

(Σημείωση: Το πρόγραμμα Dynamic_Median έχει bug και δεν τρέχει διότι σταματά στην γραμμή 68 στην 2η επανάληψη πετώντας NullPointerException. Παρόλαυτα ο αλγόριθμος που ακολουθεί έχει την λογική που έχει περιγραφεί απο πάνω.)