

קורס: מבני נתונים 1 - 234218

חלק יבש לרטוב 1

מגישות:

1. אמילי דרור [324934082]
2. עדי צח [207844358]

תאריך הגשה: 24.05.2023

**תיאור מבנה הניתונים**

מחלקת streaming\_database

עץ משתמשים [users] – עץ avl שמחזיק את המשתמשים Users במערכת, המפתחות יהיו userId.

עץ קבוצות [groups] – עץ avl שמחזיק את הקבוצות Groups במערכת, המפתחות יהיו groupId.

עץ סרטים [movies\_id] – עץ avl שמחזיק את הסרטים Movies במערכת, המפתחות יהיו movieId.

מערך של עצי סרטים לפי ז׳אנר [movies\_genre] – מערך של עצי avl בגודל מספר הז׳אנרים פלוס 1, העצים יחזיקו מצביעים לסרטים Movies \* והמפתחות יהיו הסרטים עצמם, העץ האחרון במערך זה יכיל את כל הסרטים.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת User

שדות:

* id - מזהה ייחודי
* vip - שדה בוליאני שמציין אם המשתמש vip
* views[] - מערך שמכיל מספר הצפיות בכל ז׳אנר
* group - מצביע לקבוצה שאליה המשתמש שייך (יוגדר להיות nullptr אם אינו שייך לשום קבוצה)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Group

שדות:

* id - מזהה ייחודי
* vip - שדה שמציין את מספר משתמשי vip בקבוצה
* views[] - מערך שמכיל מספר הצפיות בכל ז׳אנר
* users - עץ avl שמחזיק את מצביעים למשתמשים Users \* במערכת, המפתחות יהיו userId

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Movie

שדות:

* id - מזהה ייחודי
* vip - שדה שמציין את מספר משתמשי vip בקבוצה
* views[] - מערך שמכיל מספר הצפיות בכל ז׳אנר
* users - עץ avl שמחזיק את מצביעים למשתמשים Users \* במערכת, המפתחות יהיו userId

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Movie

*תיאורי קצר: מחלקה זו גנרית במפתחות וגם במידע שצריך להחזיק*

*שדות: מצביע לשורש, גודל העץ.*

*חיפוש, הוספה והוצאה – לפי האלגוריתמים שנלמדו בכיתה .*

*סיורים בעץ – לפי האלגוריתמים שנלמדו בכיתה .*

*איחוד שני עצים – בעזרת סיור* inorder *ניצור שני מערכים ממוינים של איברי שני העצים, נאחד את שני המערכים בעזרת פונקצית* merge *פשוטה וניעזר בפונקציה אחרת שמבצעת את ההפך מסיור* inorder *כדי ליצור עץ ונחזיר אותו, הסיבוכיות של הפתרון היא , כאשר הוא גודל שני העצים ביחד.*

*פונקציות מקסימום ומינימום – נחזיר את האיבר בעל המפתח הכי גדול או קטן על ידי החזרת העלה הימנית ביותר או השמאלית ביותר .*

**­­­­­­**

**תיאור המתודות של streaming\_database:**

streaming\_database()

*תיאור:* .

*סיבוכיות: .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

~streaming\_database()

*תיאור:* .

*סיבוכיות: .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_movie(int movieId, Genre genre, int views, bool vipOnly)

*תיאור: נייצר אובייקט סרט חדש ונכניס אותו לשלושה עצים, עץ הסרטים [id] ולשני עצים במערך עצי הסרטים, עץ הסרטים מז׳אנר ­­* genre*והעץ אחרון הבמערך. כמו כן, נוודא את נכונות הקלט, ונחזיר את ערך החזרה הדרוש בהתאם. נוסיף 1 למונה של כלל הסרטים ולמונה של הז׳אנר המתאים.*

*סיבוכיות: ייצור אובייקט הכנסה לשלוש עצי avl (נשים לב שגדלי בעצים חסומים ע״י מספר הסרטים הכולל במערכת).*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType remove\_movie(int movieId)

*תיאור: נחפש את הסרט בעץ* movies\_id *בעזרת האובייקט המוחזר נוציא את הסרט משני העצים הרלווניטים במערך העצים ואח״כ נוציא אותו מעץ* movies\_id. נחסיר *1 מהמונה של כלל הסרטים ומהמונה של הז׳אנר המתאים.*

*סיבוכיות: פעולות חיפוש והוצאה מעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_user(int userId, bool isVip)

*תיאור: נייצר אובייקט משתמש חדש ונכניס אותו לעץ* users*.*

*סיבוכיות: ייצור אובייקט הכנסה לעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType remove\_user(int userId)

*תיאור: נוציא את המשתמש בעל המספר הייחודי* userId *מעץ* users*.*

*סיבוכיות: הוצאה מעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_group(int groupId)

*תיאור: נייצר אובייקט קבוצה חדש ונכניס אותו לעץ* groups*.*

*סיבוכיות: ייצור אובייקט הכנסה לעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType remove\_group(int groupId)

*תיאור: נוציא את הקבוצה בעלת המספר הייחודי* groupId *מעץ* groups*.*

*סיבוכיות: הוצאה מעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_user\_to\_group(int userId, int groupId)

*תיאור: נחפש את הקבוצה בעלת* groupId *בעץ* groups *ונחפש את המשתמש בעל* userId *בעץ* users *ונוסיף את המשתמש לעץ של הקבוצה ונעדכן את המשתמש כך שיצביע על קבוצה זו.*

*סיבוכיות: .*

* *חיפוש בעץ* users
* *חיפוש בעץ* groups
* *הוספה לעץ* users *במחלקתGroup*
* *עדכון שדות המשתמש*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType user\_watch(int userId, int movieId)

*תיאור: נחפש את הקבוצה בעלת* groupId *בעץ* groups *ונחפש את המשתמש בעל* userId *בעץ* users, נבדוק שהמשתמש מוגדר להיות vip אם הסרט מוגדר להיות vip *ונעדכן את שדה* views *של המשתמש.*

*סיבוכיות: .*

* *חיפוש בעץ* users
* *חיפוש בעץ* groups
* *עדכון שדות המשתמש*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType group\_watch(int groupId, int movieId)

*תיאור: נחפש את הקבוצה בעלת* groupId *בעץ* groups *ונחפש את הסרט בעל* movieId *בעץ* movies,נבדוק שמספר המשתמשים שמוגדר להיות vip גדול מאפס אם הסרט מוגדר להיות vip *ונעדכן את שדה* views *של הקבוצה.*

*סיבוכיות: .*

* *חיפוש בעץ* movies
* *חיפוש בעץ* groups
* *עדכון שדות הקבוצה*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t < int > get\_all\_movies\_count(Genre genre)

*תיאור: ניגש לעץ* movies\_genre[genre] *ובעזרת מתודת* size() *נקבל את מספר הסרטים בעץ ולכן מספר הסרטים מז׳אנר הזה במערכת או את המספר הכולל.*

*סיבוכיות: מציאת העץ והפעלת מתודת* size() *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType get\_all\_movies(Genre genre, int ∗ const output)

*תיאור: ניגש לעץ* movies\_genre[genre] *ובעזרת סיור* inorder *נמלא מערך* output*.*

*סיבוכיות: סיור* inorder *אם* genre=NONE *מקיים , אחרת .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t < int > get\_num\_views(int userId, Genre genre)

*תיאור: נחפש את המשתמש בעל* userId *בעץ* users, נסכום את views[genre] שהמשתמש מחזיק ו- views[genre] שקבוצת המשתמש מחזיקה (בעזרת מצביע group של המשתמש). (כאשר מוסיפים משתמש מחסירים views[genre] ממנו)

*סיבוכיות: חיפוש בעץ* users *וסכימה לכן .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType rate\_movie(int userId, int movieId, int rating)

*תיאור: נחפש את הסרט בעלת* movieId *בעץ* movies *ונחפש את המשתמש בעל* userId *בעץ* users, נסיר אותו מהעץץץץץץץץים?, נסיף לו את הדירוג

*סיבוכיות: .*

* *חיפוש בעץ* movies
* *חיפוש בעץ* groups
* *עדכון שדות הקבוצה*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t < int > get\_group\_recommendation(int groupId)

*תיאור: נשמור שני שדות במחלקת* Player*, כאשר מכניסים, מוציאים או מעדכנים שחקן ננצל שפעולות אלה עולות ונעדכן שדות אלה לשחקן עצמו ולשני השחקנים ששדות אלה מצביעים עליהם. עכשיו נחפש את הקבוצה הנתונה ונחפש בעץ שהיא מחזיקה את השחקן, נשווה בין שני השדות ומחזיר את הקרוב ביותר.*

*סיבוכיות: חיפוש בעץ הקבוצות , חיפוש בעץ השחקנים בתוך הקבוצה , השוואה לכן .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*