

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 במערכת.

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

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
* Total\_viewes[] - מערך שמכיל מספר הצפיות בכל ז׳אנר
* parameter\_views[] - מערך שמכיל פרמטר אודות מספר הצפיות בכל ז׳אנר אשר אנשי הקבוצה נעזרים בו על מנת לחשב את סך הצפיות האישיות שלהם.
* members - עץ avl שמחזיק את מצביעים למשתמשים Users \* בקבוצה, המפתחות יהיו userId

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

מחלקת Movie

שדות:

* id - מזהה ייחודי
* vip - שדה שמציין האם הסרט vip
* views – int שמכיל מספר הצפיות בסרט
* genre – משתנה של הז׳אנר של הסרט
* tot\_points – כמות הנקודות שניתנה לסרט
* number\_of\_ratings – כמות הפעמים שהסרט הנתנון קיבל דירוג כלשהו

**תיאור המתודות של streaming\_database:**

streaming\_database()

*תיאור:* .נאתחל את כלל העצים הקיימים במערכת שלנו עם ערכי null (עץ משתמשים, עץ סרטים, עץ קבוצות ועצי הסרטים לכל ז׳אנר).

*סיבוכיות: .* O(1)

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

~streaming\_database()

*תיאור:* .נמחוק את המידע המוחזק בעץ המשתמשים, בעץ הקבוצות ובעץ הסרטים.

סיבוכיות: O(n+m+k)

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

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

*תיאור: נבדוק את מקרי הקצה שהוגדרו*. *נייצר אובייקט סרט חדש ונכניס אותו לשלושה עצים, עץ הסרטים* movies\_id *ולשני עצים במערך עצי הסרטים, עץ הסרטים מז׳אנר ­­* genre*והעץ אחרון במערך.*

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

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

StatusType remove\_movie(int movieId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו*. *נחפש את הסרט בעץ* movies\_id *ובעזרת האובייקט המוחזר נוציא את הסרט משני העצים הרלווניטים במערך העצים ואח״כ נוציא אותו מעץ* movies\_id.

*סיבוכיות: פעולות חיפוש והוצאה מעצי avl .*

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

StatusType add\_user(int userId, bool isVip)

*תיאור: נייצר אובייקט משתמש חדש ונכניס אותו לעץ* users*. נבדוק את מקרי הקצה שהוגדרו*.

*סיבוכיות: ייצור אובייקט הכנסה לעצי avl .*

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

StatusType remove\_user(int userId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו*. *נוציא את המשתמש בעל המספר הייחודי* userId *מעץ* users*. במידה והמתמש שייך לקבוצה אזי נאעדכן את השדות הרלוונטיים בקבוצה (כמות הצפיות,vip, עץ המשתמשים השייכים לקבוצה).*

*סיבוכיות: הוצאה מעצי avl .*

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

StatusType add\_group(int groupId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו* ו*נייצר אובייקט קבוצה חדש, נכניס אותו לעץ* groups*.*

*סיבוכיות: ייצור אובייקט הכנסה לעצי avl .*

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

StatusType remove\_group(int groupId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו*. *נדאג לעדכן עבור המשתמשים השייכים לקבוצה את השדות שלהם, גם את שדה הצפיות שלהם וגם את שדה השייכות לקבוצה. נוציא את הקבוצה בעלת המספר הייחודי* groupId *מעץ* groups*.*

*סיבוכיות: הוצאה מעצי avlועדכון משתמשי הקבוצה . כאשרn מתייחס אך ורק למשתמשים הנמצאים בקבוצה.*

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

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

*תיאור: נחפש את הקבוצה בעלת* groupId *בעץ* groups *ונחפש את המשתמש בעל* userId *בעץ* users *ונבדוק את מקרי הקצה שהוגדרו (לדוגמא במידה והמשתמש שייך לקבוצה אזי נחזיק failure)*. *נוסיף את המשתמש לעץ של הקבוצה, נעדכן את המשתמש כך שיצביע על קבוצה זו, נוסיף לשדות הקבוצה את צפיות המשתמש ובמידה והינו משתמש vip נשנה את השדה הרלוונטי בהתאם ונעדכן את ערך צפיות המשתמש כך שסכימה שלו עם הפרמטר צפיות בקבוצה יתן את הערך הרצוי.*

*סיבוכיות: .*

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

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

StatusType user\_watch(int userId, int movieId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו* ו*נחפש את הסרט בעל* movieId *בעץ* movies\_id *ונחפש את המשתמש בעל* userId *בעץ* users, נבדוק שהמשתמש מוגדר להיות vip אם הסרט מוגדר להיות vip *ונעדכן את שדה* views *של המשתמש. במידה והמשתמש שייך לקבוצה אזי נעדכן את שדה הצפיות הכלליות בקבוצה. נסיר ונוסיף סרט משני העתים שלו הממוינים לפי דירוג.*

*סיבוכיות: .*

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

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

StatusType group\_watch(int groupId, int movieId)

*תיאור: נבדוק את מקרי הקצה שהוגדרו, 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. . נסיר ונוסיף סרט משני העתים שלו הממוינים לפי דירוג.*

*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*. כמו כן לאורך ביצוע הפעולות הנ״ל נבדוק את מקרי הקצה שהוגדרו ונחזיר את הערך הרלוונטי.*

*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 ובמידה ושייך לקבוצה גם views[genre] שקבוצת המשתמש מחזיקה (בעזרת מצביע group של המשתמש). (כאשר מוסיפים משתמש מחסירים views[genre] ממנו) *כמו כן לאורך ביצוע הפעולות הנ״ל נבדוק את מקרי הקצה שהוגדרו ונחזיר את הערך הרלוונטי.*

*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, נבדוק כי מבחינת דרישות vip ניתן לבצע את הפעולה. נסיר את הסרט משני העצים הממוינים על פי דירוג הסרטים בהם הוא נמצא. נעדכן את שדות הסרט מבחינת נקודות וכמות דירוגים ונחזיר את הסרט לשני העצים בהם היה. *כמו כן לאורך ביצוע הפעולות הנ״ל נבדוק את מקרי הקצה שהוגדרו ונחזיר את הערך הרלוונטי.*

*סיבוכיות: .*

* *חיפוש והסרה בעצי* movies
* *חיפוש בעץ* המשתמשים
* *עדכון שדות הקבוצה*

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

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

*תיאור: 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 .*

*לא עומדים כרגע בסיבוכיות המתורת לפונקציה זו צריך לשמור את הסרט המקסימלי בכל עץ!!! בכל הכנסה לעצים הממוינים דירוג לבדוק האם גדול יותר ממקסימלי.*

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