מבני נתונים 234218 תרגיל רטוב 1, חלק יבש

תקציר:

מסמך זה מתאר את מבנה הפתרון של התרגיל. מבנה הנתונים העיקרי בו השתמשנו הוא עץ AVL. בחלקו הראשון של המסמך נתאר את המחלקות והפונקציות הבסיסיות בהן אנו משתמשים ונוכיח את הסיבוכיות שלהן. בחלק השני של המסמך נוכיח עבור כל אחת מהמתודות שנתבקשנו לממש כי היא אכן עומדת בדרישות הסיבוכיות

חלק -1 מבני הנתונים בהם בחרנו להשתמש :

1) עצים ממוינים AVL
2) מערכים רגילים

שימוש בעץ :

עץ זה הוא מקרה פרטי של עץ בינארי. מכיל node כאשר לכל node יכול להיות הורה, בן שמאלי או בן ימני.

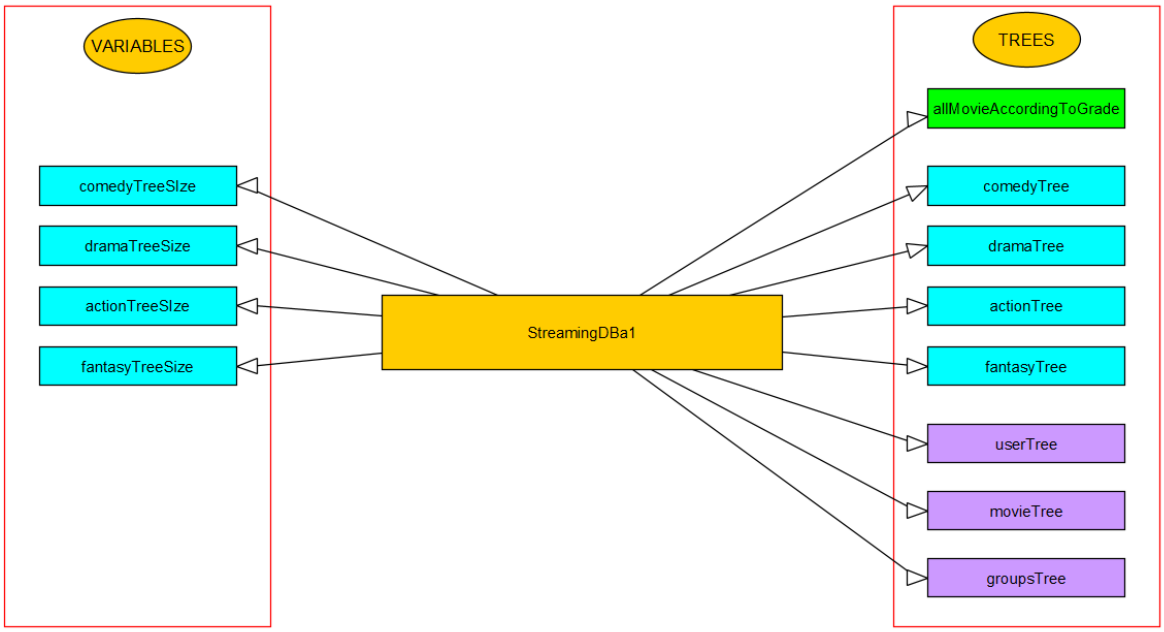
העץ ממומש באופן גנרי באמצעות template החיפוש, ההוצאה וההכנסה מומשו לפי האלגוריתמים שנלמדו בהרצאה. – בהרצאה הוכח שבעץ AVL כזה חיפוש, הכנסה והוצאה של node מתבצע בסיבוכיות זמן של $O(\log(n))$ כמו כן הוכח שבעץ AVL עם גובה h ו n קודקודים מתקיים $O(h) = O(\log(n))$ כל מופע של עץ מצביע ל node שמהווה את ה root נשים לב כי root לא קבוע ויכול להשתנות עם הוצאה והכנסה של node

Part 2 : The implementation :

The different classes :

1- StreamingDBa1, the class that runs the whole project, she contains 8 trees:
   a- usersTree : a tree of all users sort according to the ID.
   b- groupsTree : a tree of all groups sort according to the ID.
   c- moviesTree : a tree of all movies sort according to the ID.
   d- comedyTree , actionTree, dramaTree , fantasyTree : 4 tree of the movies of each genre sort according to their average.
   e- allMovieAccordingToGrade : a tree of all movies sort according to their average.

We used smart pointers because we wanted a more convenient way to manage memory, and the reason we specifically used shared pointers instead of unique pointers is that we needed multiple objects to hold a pointer to the same object. For example, the user's tree holds a pointer to the user, but also the user tree of each group can hold a pointer to the same user.



2- User : The class is representing a user. The class contain:
   a- the ID of the user;
   b- A Boolean to know if he is user VIP;
   c- The ID of his group (0 if he is not doing part of a group);
   d- a pointer to his group;
   e- a list with the number of movie of each genre that he saw;
   f- a list with the number of movie of each genre that his group saw before his entry in the group;

3- Movie: The class is representing a movie. The class contain:
   a- The ID of the movie;
   b- The genre of the movie;
   c- The number of views of the movie;
   d- A Boolean to know if he is just for the user VIP;
   e- The sum of his rating;
   f- The number of user that rate him;

4- Group: the class is representing a group. The class contain :
   a- The ID of the group;
   b- How many users there is in the group;
   c- The number of user VIP in the group that we will update everytime that a user join or leave the group;
   d- A list of all of the movie seen by the group and the user of the group in every moment according to the genre. For example if a group of 3 users see an action movie, we add to the cell “action” **3**. And if a user of the group see a movie alone we also add 1 to this list;
   e- A list of the number of movie seen by the group and only the group. For example if a group of 3 users see an action movie, we add to the cell “action” **1**.
   f- A pointer to the usersTree.

The methods of streamingDBa1 :

1) add_movie

הפונקציה מקבלת ID של סרט ואת הפרמטרים הנחוצים לסרט
ראשית נבדוק האם הקלט תקין והאם לא קיים סרט עם ID זהה לזה של החדש, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם.

ברגע שנעבור את הבדיקות נכניס את הסרט לעץ הסרטים וגם נוסיף אותו לעץ כל הז'אנרים
נוסיף אותו גם לעץ הז'אנר הספציפי ונעדכן את הסרט עם הדירוג הגבוה ביותר בז'אנר של הסרט שראינו במידה והשתנה משהו

Complexity	Action
$O(logk)$	חיפוש הסרט בעץ הסרטים
$O(logk)$	הוספה לעץ הסרטים
$O(logk + logk)$	הוספה לעץ כל הז'אנרים ולעץ הז'אנר הספציפי
$O(logk)$	חיפוש הסרט המוביל בז'אנר ספציפי

$$\boldsymbol{Total = O(logk)}$$

2) remove_movie

הפונקציה מקבלת ID של סרט
ראשית נבדוק האם הקלט תקין והאם קיים סרט עם ID כזה

אם אכן תקין, נוציא את הסרט מהעץ של הז'אנר הספציפי וגם מעץ כל הז'אנרים וגם מעץ הסרטים
נעדכן בהתאם את הסרט המוביל בז'אנר הספציפי במידה והוא היה הסרט המדורג הכי גבוה, ובמידה וכרגע אין בכלל סרטים בז'אנר אז נקבע שאין סרט מוביל בכלל.

Complexity	Action
$O(logk)$	חיפוש הסרט בעץ הסרטים
$O(logk)$	מחיקה מעץ הסרטים
$O(logk + logk)$	מחיקה מעץ הז'אנרים ועץ הז'אנר הספציפי
$O(logk)$	חיפוש הסרט המוביל בז'אנר ספציפי

$$\boldsymbol{Total = O(logn)}$$

3) add_user

הפונקציה מקבלת ID של משתמש
ראשית נבדוק האם הקלט תקין והאם לא קיים משתמש עם ID זהה לזה של החדש, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם.

ברגע שנעבור את הבדיקות נכניס את המשתמש לעץ המשתמשים

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(logn)$	הוספה לעץ המשתמשים

$$\boldsymbol{Total = O(logn)}$$

4) remove_user

הפונקציה מקבלת ID של משתמש
ראשית נבדוק האם הקלט תקין והאם המשתמש בכלל קיים, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם.

אם המשתמש חלק מקבוצה , ניגש לקבוצה ונסיר את המשתמש לפי המתודה של הקבוצה שמסירה את המשתמש ומעדכנת את המידע שלה בהתאם

לאחר מכן נסיר את המשתמש מעץ המשתמשים

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(logn)$	מחיקה מעץ המשתמשים

$$\boldsymbol{Total = O(logn)}$$

5) add_group

הפונקציה מקבלת ID של קבוצה
ראשית נבדוק האם הקלט תקין והאם הקבוצה כבר קיימת, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם.

ברגע שנעבור את הבדיקות נכניס את הקבוצה לעץ הקבוצות כאשר סטטוס הVIP שלה הוא false

Complexity	Action
$O(logm)$	חיפוש הקבוצה בעץ הקבוצות
$O(logm)$	הוספה לעץ הקבוצות

$$\boldsymbol{Total = O(logm)}$$

6) remove_group

הפונקציה מקבלת ID של קבוצה
ראשית נבדוק האם הקלט תקין והאם הקבוצה לא קיימת, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם.

נמחק את כל המשתמשים מהקבוצה בעזרת המתודה של הקבוצה removeMe, שמעדכנת לכל משתמש בקבוצה את כמות הצפיות שהייתה לקבוצה איתו כדי שהצפיות שלו יסתנכרנו בהתאם וגם נגדיר עבורו שכרגע אין לו קבוצה

ברגע שסיימנו למחוק את כל המשתמשים נמחק את הקבוצה מעץ הקבוצות

Complexity	Action
$O(logm)$	חיפוש הקבוצה בעץ הקבוצות
$O(logm)$	מחיקת הקבוצה מעץ הקבוצות
$O(n_{groupUsers})$	עדכון הצפיות של כל אחד מהמשתמשים ועדכון שהם כבר לא חלק מקבוצה (לכל היותר n משתמשים בקבוצה)

$$\boldsymbol{Total = O\left(logm + n_{groupUsers}\right)}$$

add_user_to_group (7

הפונקציה מקבלת ID של משתמש וID של קבוצה
ראשית נבדוק האם הקלט תקין והאם המשתמש כבר קיים והקבוצה קיימת וגם שהמשתמש לא שייך לקבוצה אחרת, בכל אחד מהמקרים האלה נחזיר שגיאה בהתאם

ברגע שנעבור את הבדיקות נשתמש במתודה של הקבוצה להכנסת משתמש, שמעדכנת את הצפיות המשותפות של הקבוצה בהתאם לצפיות של המשתמש, וגם מעדכנת את סטטוס הVIP שלה במידה ויש צורך, ולבסוף גם ניגש אל המשתמש ונגדיר עבורו שהוא חלק מקבוצה עם כל מה שמלווה לזה

נוסיף את המשתמש אל עץ המשתמשים של הקבוצה

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ
$O(logn)$	הוספה לעץ המשתמשים
$O(logm)$	חיפוש הקבוצה בעץ הקבוצות
$O(logm)$	הוספה לעץ הקבוצות

$$\boldsymbol{Total = O(logn + logm)}$$

user_watch (8

הפונקציה מקבלת ID של משתמש וID של סרט
ראשית נבדוק האם הקלט תקין והאם המשתמש כבר קיים והסרט קיים

אם המשתמש לא VIP והסרט כן VIP אז לא נוכל לצפות בסרט

נוסיף לצפיות הז'אנרים של המשתמש את הצפיה בז'אנר של הסרט הנוכחי וגם במידה והוא חלק מקבוצה נעדכן את הצפיות הכוללות שלה בז'אנר

ניגש גם לעץ הז'אנר וגם לעץ של כל הז'אנרים ונעדכן את הצפיות בסרט
נעדכן את הסרט עם הדירוג הגבוה ביותר בז'אנר של הסרט שראינו במידה והשתנה משהו

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(logk)$	חיפוש הסרט בעץ הסרטים
$O(logk + logk)$	עדכון עץ כל הסרטים ועץ הז'אנר הספציפי
$O(1)$	גישה ועדכון הקבוצה
$O(logk)$	חיפוש הסרט המוביל בז'אנר ספציפי

$$\boldsymbol{Total = O(logn + logk)}$$

group_watch (9

הפונקציה מקבלת ID של קבוצה וID של סרט
ראשית נבדוק האם הקלט תקין והאם הקבוצה כבר קיימת והסרט קיים

אם הקבוצה לא VIP והסרט כן VIP אז לא נוכל לצפות בסרט

נוסיף לצפיות בז'אנר של הקבוצה צפיות ככמות המשתמשים שנמצאים בקבוצה במידה ונמצאים כאלה וגם נוסיף צפייה משותפת אחת כדי שכל אחד מהמשתמשים יוכל לדעת שיש לו עוד צפייה.

ניגש גם לעץ הז'אנר וגם לעץ של כל הז'אנרים ונעדכן את הצפיות בסרט
נעדכן את הסרט עם הדירוג הגבוה ביותר בז'אנר של הסרט שראינו במידה והשתנה משהו

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(logk)$	חיפוש הסרט בעץ הסרטים
$O(logk + logk)$	עדכון עץ כל הסרטים ועץ הז'אנר הספציפי
$O(1)$	גישה ועדכון הקבוצה
$O(logk)$	חיפוש הסרט המוביל בז'אנר ספציפי

$$\boldsymbol{Total = O(logn + logk)}$$

<u>get_num_views (12</u>

הפונקציה מקבלת ID של מתשתמש וז'אנר כלשהו
ראשית נבדוק האם הקלט תקין והאם המשתמש קיים

אם למשתמש אין קבוצה נחזיר בדיוק את כמות הצפיות שלו בז'אנר
אם למשתמש יש קבוצה נחשב את כמות הצפיות שלו ונוסיף לזה את כמות הצפיות של הקבוצה
פחות כמות הצפיות שהיו לקבוצה לפני שנכנס וכך נקבל את מס' הצפיות המדויק שלו בז'אנר

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(1)$	חישוב הצפיות
$O(1)$	גישה לקבוצה

$$\boldsymbol{Total = O(logn)}$$

<u>rate_movie (13</u>

הפונקציה מקבלת ID של משתמש, ID של סרט ודירוג כלשהו.
ראשית נבדוק האם הקלט תקין מבחינת המשתמש הדירוג והסרט.

אם תקין, נוודא שבמידה והסרט VIP אז גם המשתמש VIP אחרת לא יוכל לדרגו.

כעת בהתאם לז'אנר של הסרט נעדכן את העץ ז'אנר המתאים וגם את את כל הז'אנרים (לפי דירוג לא לפי ID) עם הדירוג החדש של הסרט
נעדכן את הסרט עם הדירוג הגבוה ביותר בז'אנר של הסרט שראינו במידה והשתנה משהו

Complexity	Action
$O(logn)$	חיפוש המשתמש בעץ המשתמשים
$O(logk)$	חיפוש הסרט בעץ הסרטים
$O(logk + logk)$	עדכון עץ כל הסרטים ועץ הז'אנר הספציפי
$O(logk)$	חיפוש הסרט המוביל בז'אנר ספציפי

$$\boldsymbol{Total = O(logn + logk)}$$

get_group_recommendation (14

הפונקציה מקבלת ID של קבוצה
ראשית נבדוק האם הקלט תקין והאם הקבוצה כבר קיימת

נבדוק במערך הצפיות כולל המשתמשים של הקבוצה מהו הז'אנר הכי נצפה
בהינתן שהז'אנר אינו NONE, ניגש לסרט עם הדירוג הכי גבוה בעץ של הז'אנר הנוכחי ונחזיר את הID של הסרט הזה

Complexity	Action
$O(logm)$	חיפוש הקבוצה בעץ הקבוצות
$O(1)$	מציאת הז'אנר הכי נצפה
$O(1)$	החזרת הסרט הכי נצפה לפי הז'אנר

$$\boldsymbol{Total = O(logm)}$$

get_all_movie_count (15

הפונקציה_ מקבלת ג"אר ומחזירה את המשתנה שמתאים.

Complexity	Action
$O(1)$	להחזיר משתנה

$$\boldsymbol{Total = O(1)}$$

get_all_movie (16

הפונקציה מקבלת ג"אר ומצביע לתחילת מערך.

ראשית נבדוק האם הקלט תקין והאם הג"אר לא ריק

הפונקציה בודקת מה הג"אר הנתון ולפי זה הולכת בעץ של הג"אר המתאים בסיבוכיות של ואז בזכות הפונקציה הוא מחזיר את האיברים של העץ. ואז אנחנו הופכים את הרשימה כדי שיהיה בסדר המבוקש בסיבוכיות של
1 $O(Kgenre)$

Complexity	Action
$O(1)$	מציאת העץ המתאים
$O(Kgenre)$	מחזירים כל איבר של העץ ברשימה
$O(Kgenre)$	הופכים את הרשימה

$$\boldsymbol{Total = O(Kgenre)}$$