

רטוב 1:

הסבר על מבני הנתונים הבסיסיים שהשתמשנו בהם :

המבנה הבסיסי:

עץ AVL :

בסיבוכיות זמן $\log(n)$ השימוש העיקרי במבנה הזה הוא על מנת לנצל את התכונה של חיפוש הזנה והסרת איברים בעץ AVL של

מימשנו מילון גנרי בעל מפתח וערך גנריים באמצעות עץ ה-AVL, נשתמש בהם בטיפוסים שונים לטובת המימוש שלנו.

המבנה העיקרי :

כולל שני עצים עיקריים :

1. עץ של קבוצות:

כל מפתח בעץ הוא המספר המזהה של החברה (ID) (העץ ממוין לפי מספר מזהה), וזה data של כל קבוצה הוא טיפוס בשם GROUP שכולל את id של הקבוצה, מספר המשתמשים שהם VIP, מספר הסרטים שצפו בהם מכל סוג כקבוצה והג'אנר בעל הצפיות המקסימלי בקבוצה, עץ של המשתמשים ששייכים לקבוצה לכל משתמש בעץ זה המפתח -ID של המשתמש ו-DATA הוא טיפוס שכולל אם המשתמש הוא Vip, מצביע לקבוצה שלו, מספר הסרטים שצפה בהם מכל סוג.

2. עץ של משתמשים :

העץ הזה כולל כל המשתמשים שיש בכל המערכת המפתח לכל איבר בעץ הוא ID של כל המשתמש (העץ ממוין לפי מספר מזהה), ו-DATA הוא טיפוס שכולל אם המשתמש הוא Vip, מצביע לקבוצה שלו, מספר הסרטים שצפה בהם מכל סוג.

3. עץ של סרטים :

מכיל את כל הסרטים במערכת, כל מפתח בעץ הוא המספר המזהה של הסרט (ID) הטיפוס MOVIE המכיל המספר המזהה של הסרט (העץ ממוין לפי מספר מזהה), הג'אנר של הסרט, מספר הצפיות בסרט, שדה בוליאני המציין אם הסרט VIP או לא וה-RATING של הסרט, בנוסף העץ יחזיק שדה שסופר את מספר האיברים בעץ כלומר מספר הסרטים במערכת.

4. עץ של כל ג'ינר :

כל עץ מכיל את כל הסרטים במערכת השייכים לג'אנר זה, כל עץ ממוין במיון ראשי לפי שדה ה-RATING של כל סרט, מיון משני לפי הצפיות של כל סרט ומיון אחרון לפי המזהה בסדר יורד של כל סרט. מכיל הטיפוס MOVIE המכיל המספר המזהה של הסרט, הג'אנר של הסרט, מספר הצפיות בסרט, שדה בוליאני המציין אם הסרט VIP או לא וה-RATING של הסרט.

בנוסף העץ יחזיק שדה שסופר את מספר האיברים בעץ כלומר מספר הסרטים בג'אנר זה במערכת וגם את המקסימום בעץ. בכל הכנסה והוצאה מעדכנים את כל השדות בעץ בהתאם.

- כולל עץ שבעצם מכיל את כל הסרטים במערכת הממוינים לפי ההסבר הנ"ל. נציין עץ זה בהסברי הפונקציות כעץ כל הג'אנרים

נפרט עבור כל פונקציה את הסיבוכיות של המימוש שלה :

**** כל הבדיקות של תקינות הקלט נעשית בסיבוכיות $O(1)$ לכן לא נזכיר אותם בניתוח הסיבוכיות בפונקציות, כנ"ל לגבי עדכון מצביעים ואובייקטים בפונקציות ****

**** בכל ביקור בכל עץ במערכת נעדכן שדה מספר האיברים בעץ בכל העצים שעשינו שינוי(הוספה או הסרה בהם) בהתאם.**

: Init

אתחול של מצביע למבנה ריק שכולל עצים ריקים לכן זה $O(1)$.

:virtual ~streaming_database()

מאחר והקצנו $O(k + n + m)$ מקום, עלינו לעבור על כל המקום כדי לשחרר אותו. לכן סיבוכיות הזמן היא בהתאם $O(k + m + n)$.

StatusType add_movie(int movieId, Genre genre, int views, bool vipOnly)

תחילה נצור אובייקט של MOVIE עם כל הפרמטרים הנתונים – $O(1)$

נבדוק אם האובייקט שיצרנו נמצא בעץ הסרטים – $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (חיפוש בעץ AVL).

אם כן נמצא נמחק את האובייקט ונחזיר שגיאה – $O(1)$

אם לא נמצא אז נוסיף אותו – $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (הוספת איבר בעץ AVL).

נוסיף אותו לעץ של הג'אנר שלו ולעץ כל הג'אנרים – $O(\log(\text{numofmoviesingenretree}))$

**** נציין שמספר הסרטים בכל ג'אנר קטן שווה ממספר הסרטים בכל המערכת לכן :

$$O(\log(\text{numofmoviesingenretree})) \leq O(\log K)$$

בסה"כ זה $O(\log k)$ במקרה הגרוע, כאשר k הוא מספר הסרטים במערכת .

StatusType remove_movie(int movieId)

מחפשים את האיבר שיש להסיר בעץ של הסרטים – $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (חיפוש בעץ AVL).

אם לא נמצא נחזיר שגיאה – $O(1)$

אחרת מוחקים את האיבר מעץ הסרטים הממוין לפי ID - $O(\log K)$

ומוחקים את האיבר מעץ הממוין בעדיפות ראשונה לפי rating והעדפות שניה views ובהעדפות שלשית לפי ID - $O(\log K)$ (עץ כל הג'אנרים)

ומוחקים את האיבר מעץ הג'אנר השייך אליו - $O(\log K)$ (ציינו למה בפונקציה הקודמת)

בסה"כ זה $O(\log k)$ במקרה הגרוע, כאשר k הוא מספר הסרטים במערכת.

StatusType add_user(int userId, bool isVip)

תחילה נצור אובייקט של USER ריק עם הפרמטרים הנתונים - אתחול מצביעם וערכים $O(1)$

נבדוק אם המשתמש קיים במערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL).

אם כן נמצא נמחק את האובייקט ונחזיר שגיאה - $O(1)$

אם לא נמצא אז נוסיף אותו - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (הוספת איבר בעץ AVL).

StatusType remove_user(int userId)

מחפשים את האיבר שיש להסיר בעץ של המשתמשים - $O(\log N)$ כאשר N מספר האיברים בעץ העובדים (חיפוש בעץ AVL).

אם לא נמצא נחזיר שגיאה - $O(1)$

אם יש מצביע לקבוצה ניגשים לעץ של הקבוצות השייכים דרך המצביע שמחזיק המשתמש בעץ המשתמשים לחברה שלו - $O(1)$ (כי לא עשינו חיפוש לקבוצה בעץ אלא שמרנו מצביע ונגשנו באמצעותו)

מוחקים את האיבר מעץ המשתמשים של הקבוצה ומעדכנים numVipUsers וגם השדות של מספר הסרטים שצפו בהם מכל סוג כקבוצה והג'אנר בעל הצפיות המקסימלי בקבוצה במידת הצורך $(\log N)$

עדכון בשדות הרלוונטיות של הקבוצה ועדכון שדה הג'אנר המקסימלי של הקבוצה - $O(1)$

מוחקים את האיבר מעץ המשתמשים של כל המערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים
סכ"ה:

$$O(\log N) + O(\log N) + O(1) + O(1) + O(\log N) \leq O(\log N)$$

במקרה הגרוע, כאשר N הוא מספר כל המשתמשים.

StatusType add_group(int groupId)

תחילה נצור אובייקט של GROUP עם כל הפרמטרים הנתונים - $O(1)$

נבדוק אם האובייקט שיצרנו נמצא בעץ הקבוצות - $O(\log m)$ כאשר m מספר האיברים בעץ הקבוצות (חיפוש בעץ AVL).

אם כן נמצא נמחק את האובייקט ונחזיר שגיאה $O(1)$.

אם לא נמצא אז נוסיף אותו $O(\log m)$ כאשר m מספר האיברים בעץ הקבוצות (הוספת איבר בעץ AVL).
בסה"כ הוספת איבר לעץ, כפי שלמדנו זה $O(\log m)$ במקרה הגרוע, כאשר m הוא מספר האיברים בעץ.

StatusType remove_group(int groupId)

מחפשים את האיבר שיש להסיר בעץ של הקבוצות $O(\log m)$ כאשר m מספר האיברים בעץ (חיפוש בעץ AVL).

אם לא נמצא נחזיר שגיאה $O(1)$.

אחרת נעבור איבר איבר בעץ המשתמשים של הקבוצה נקפוצ דרך המצביע לעץ המשתמשים העיקרי נעדכן שדה GROUP כ NULL, $O(\log(\text{numofgroupusers}))$.

מוחקים את הקבוצה מעץ הקבוצות $O(\log m)$.

סכ"ה:

$$O(\log m) + O(\log m) + O(\log(\text{numofgroupusers})) + O(1) \leq O(\log m) + O(\log(\text{numofgroupusers}))$$

במקרה הגרוע, כאשר m הוא מספר כל הקבוצות במערכת.

StatusType add_user_to_group(int userId, int groupId)

נבדוק אם הקבוצה קיימת במערכת - $O(\log m)$ כאשר m מספר האיברים בעץ הקבוצות (חיפוש בעץ AVL).

נבדוק אם המשתמש קיים במערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL).

אם הקבוצה או המשתמש לא נמצאים נחזיר שגיאה $O(1)$.

נבדוק אם המשתמש שייך לקבוצה - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL).

אם כן נמצא נחזיר שגיאה $O(1)$.

אחרת:

נוסיף את המשתמש לעץ המשתמשים השייכים לקבוצה - $O(\log(\text{numofgroupusers}))$.

***נציין שמספר המשתמשים בכל קבוצה קטן שווה ממספר המשתמשים בכל המערכת לכן:

$$O(\log N) \Rightarrow O(\log(\text{numofgroupusers}))$$

נעדכן שדה הקבוצה בעץ למשתמש בנתון בעץ המשתמשים הכללי במערכת שיצביע על הקבוצה שלו בעץ הקבוצות ושדה הצפיות - $O(\log m) + O(\log N)$.

סכ"ה:

$$O(\log m) + O(\log N) + O(\log(\text{numofgroupusers})) + O(1) \leq O(\log m) + O(\log N)$$

במקרה הגרוע, כאשר N הוא מספר כל המשתמשים במערכת ו- m הוא מספר כל הקבוצות במערכת.

StatusType user_watch(int userId, int movieId)

נבדוק אם הסרט קיים במערכת - $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (חיפוש בעץ AVL).

נבדוק אם המשתמש קיים במערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL).

אם הסרט או המשתמש לא נמצאים נחזיר שגיאה - $O(1)$

אם הסרט VIP והמשתמש לא נחזיר שגיאה - $O(1)$

אחרת : עדכון שדות של המשתמש בעץ המשתמשים : מספר הסרטים שצפה בהם בסוג של הסרט, ובשדות של הסרט בעץ הסרטים : מספר הצפיות בסרט.

אם הסרט שייך לקבוצה אז נעדכן השדות המתאימים בקבוצה שלו דרך המצביע שהוא שומר לקבוצה שלו - $O(1)$

לבסוף עדכון האיבר ששייך לסרט הנתון בעץ הג'אנר השייך אליו - $O(\log K)$

סכ"ה:

$$O(\log K) + O(\log N)$$

במקרה הגרוע, כאשר N הוא מספר כל המשתמשים במערכת ו- K הוא מספר כל הסרטים במערכת.

StatusType group_watch(int groupId, int movieId)

נבדוק אם הסרט קיים במערכת - $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (חיפוש בעץ AVL).

נבדוק אם הקבוצה קיימת במערכת - $O(\log m)$ כאשר m מספר האיברים בעץ הקבוצות (חיפוש בעץ AVL).

אם הסרט או הקבוצה לא נמצאים נחזיר שגיאה - $O(1)$

אם הסרט VIP והקבוצה לא מכילה משתמשי VIP נחזיר שגיאה - $O(1)$

אחרת : עדכון שדות של הקבוצה בעץ הקבוצות : מספר הסרטים שצפו בהם בסוג של הסרט, ובשדות של הסרט בעץ הסרטים : מספר הצפיות בסרט.

לבסוף עדכון האיבר ששייך לסרט הנתון בעץ הג'אנר השייך אליו - $O(\log K)$

סכ"ה:

$$O(\log K) + O(\log m)$$

במקרה הגרוע, כאשר m הוא מספר כל הקבוצות במערכת ו- K הוא מספר כל הסרטים במערכת.

output t < int > get_all_movies_count(Genre genre)

אם $NONE=genre$, מחזירים הערך שסופר מספר האיברים בעץ של כל הסרטים במערכת - $O(1)$

אחרת מחזירים הערך שסופר מספר האיברים בעץ של כל הסרטים בג'אנר הנתון - $O(1)$

StatusType get_all_movies(Genre genre, int * const output)

אם $NONE=genre$, נחזיק גודל העץ הממוין בעדיפות ראשונה לפי rating והעדפות שניה views ובהעדפות שלשית לפי ID של כל הסרטים - $O(1)$, נבצע על העץ הזה pre-order ונכניס ונכניס האברים מסוף המערך להתחלה בעזרת המשתנה שמחזיק גודל העץ ונקטין אותו אחרי כל הכנסה ב- $O(k)$.

אם הערך שמוחזר של שסופר מספר האיברים בעץ של כל הסרטים העץ הנתון אפס נחזיר שגיאה - $O(1)$

. סכ"ה:

$$O(\log K) + O(1) = O(\log K)$$

K מספר מספר האברים המכיל העץ הזה והם מספר הסרטים במערכת.

אחרת, נחזיק הערך שסופר מספר האיברים בעץ של כל הסרטים בג'נר הנתון - $O(1)$, נבצע על העץ הזה (עץ הג'אנר) pre-order ונכניס הברים מסוף המערך להתחלה בעזרת המשתנה שמחזיק גודל העץ ונקטין אותו אחרי כל הכנסה ב- $O(k_{\text{gener}})$.

k_{gener} עץ הג'אנר יכיל אברים מהג'אנר המסוים.

אם הערך שמוחזר של שסופר מספר האיברים בעץ של כל הסרטים בג'נר הנתון אפס נחזיר שגיאה - $O(1)$

. סכ"ה:

$$O(\log K_{\text{gener}}) + O(1) = O(\log K_{\text{gener}})$$

output t < int > get_num_views(int userId, Genre genre)

נבדוק אם המשתמש קיים במערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL) אם קיים נחזיק עותק ממנו.

אם המשתמש לא נמצא נחזיר שגיאה - $O(1)$.

(1) אם $genre = NONE$, ניקש דרך העותק שמחזקים למצביע על קבוצה אם הוא שווה `nullptr` אז המשתמש אנו שייך לקבוצה אז נסכם השדות שמצגות כמה סרטים צפה מכל סוג מסוגים השונים, אם המצביע אנו שווה `nullptr` אז הוא מצביע על קבוצה אז שסומכים כל סוג נוסיף כמה צפתה הקבוצה מסוג הזה דרך הגשה לשדה המתים דרך המצביע שמצביע על הקבוצה בתוך המשתמש ב - $O(1)$.

דאגנו שאספנו היוזר לקבוצה להוריד הצפיות של כל סוג מצפיות של המשתמש משום שנסכם בסוף הצפיות של הקבוצה לפני שנוסיף אותו יתגזזו.

הסכמה מתבצעת ב - $O(1)$.

(2) אחרת אם היוזר לא שייך לקבוצה נחזיר מספר הצפיות מסוג מסוים, אם הוא שייך לקבוצה נחזיר מספר הצפיות מסוג מסוים סכום הצפיות של הקבוצה מסוג הזה ב - $O(1)$.
התהקות עם איך לקשת לקבוצה קמו 1.

. סכ"ה:

$$O(\log N) + O(1) = O(\log N)$$

StatusType rate_movie(int userId, int movieId, int rating)

נבדוק אם הסרט קיים במערכת - $O(\log K)$ כאשר K מספר האיברים בעץ הסרטים (חיפוש בעץ AVL).

נבדוק אם המשתמש קיים במערכת - $O(\log N)$ כאשר N מספר האיברים בעץ המשתמשים (חיפוש בעץ AVL).

אם הסרט או המשתמש לא נמצא נחזיר שגיאה - $O(1)$.

אם הסרט VIP והמשתמש לא נחזיר שגיאה - $O(1)$.

ניגש לשדה ה-RATING של הסרט המבוקש בעץ הסרטים ונעדכן אותו - $O(1) + O(\log K)$.

באותו אופן בעץ הג'אנר השייך אליו הסרט $O(1) + O(\log K)$

סכ"ה:

$$O(\log K) + O(\log N)$$

במקרה הגרוע, כאשר N הוא מספר כל המשתמשים במערכת ו- K הוא מספר כל הסרטים במערכת.

output t < int > get_group_recommendation(int groupId)

נבדוק אם הקבוצה קיימת במערכת - $O(\log m)$ כאשר m מספר האיברים בעץ הקבוצות (חיפוש בעץ AVL).

אם הקבוצה לא קיימת נחזיר שגיאה - $O(1)$

נבדוק אם הקבוצה מכילה משתמשים (עץ המשתמשים בקבוצה שומר כמות המשתמשים בעץ) - $O(1)$

אם הקבוצה ריקה נחזיר שגיאה - $O(1)$

אחרת : ניגשים לשדה ששומר את הג'אנר בעל צפיות המקסימלי - $O(1)$

יש לנו עכשיו מידע מה הג'אנר הכי מבוקש, נעבור לעץ של הג'אנר הזה :

נבדוק אם העץ של הג'אנר ריק נחזיר שגיאה - $O(1)$

אחרת נחזיר הערך ששומר בשדה המקסימום בעץ - $O(1)$

סכ"ה: $O(\log m)$ במקרה הגרוע, כאשר m הוא מספר כל הקבוצות במערכת.