רטוב 1 - חלק יבש

22:15 Sunday, 14 May 2023

נממש את כל ההעצים בתרגיל בעזרת עצי AVL. עצי AVL כפי שהוכח בהרצאות הם עצים בינאריים המשמרים על סיבוכיות זמן של פעולות הוצאה, הכנסה ומציאת איברים בעץ של $O(\log n)$ וסיבוכיות מקום של $O(n)$ כלומר נוכל באופן חופשי לבצע פעולות אלה על העץ בלי לדאוג שהסיבוכיות של העץ תחרוג מזה.

נבצע את השינויים הבאים למבנה העץ AVL שלנו

1. לכל צומת בעץ יהיה מצביע לצומת ההורה שלו - זה מוסיף סך הכל מצביע אחד לכל צומת, כלומר n מצביעים ולא משנה את סיבוכיות הזיכרון של העץ. ההשפעה היחידה של שדה זה על מימושם של הפעולות של העץ (הכנסה, הוצאה, מציאה) היא שכל פעם שנרצה להפוך צומת אחד לילד של צומת אחר נצטרך בנוסף לשנות את שדה ההורה של הילד. כלומר אנחנו לכל היותר מכפילים פי 2 את הסיבוכיות של כל פעולה כזאתי משמע זה לא משנה את הסיבוכיות הזמן של ביצוע הפעולות.
2. העץ ישמור את מספר הצמתים בעץ - זה int אחד ולכן לא משפיע על סיבוכיות המקום של העץ. בכל הוצאה של איבר מהעץ המספר יקטן ב1 ובכל הכנסה הוא יגדל ב1. שתי הפעולות האלה הן $O(1)$ ולא משפיעות על המבנה של העץ ולכן לא משנות את הסיבוכיות של העץ.
3. עץ יוכל לכלול מפתחות מטיפוסים שונים - המפתח של העץ לא יהיה בהכרח מספר אלא יוכל להיות גם כל טיפוס אחר, כל עוד זה טיפוס שיש לו סדר ושניתן להשוות בינו לבין איבר אחר ב$O(1)$. כיוון שפעולת ההשוואה נשארת $O(1)$ זה לא משנה את הסיבוכיות של אף פעולה בעץ וכל עוד כמות הזיכרון שמפתח תופס הוא חסום זה לכל היותר מגדיל את הזיכרון ב$O(n)$ ולא חורג מהמגבלות של העץ.

חוץ מהשינויים האלה המימוש זהה למימוש הסטנדרטי שהועבר בשיעור ולכן לא יפורט כאן.

n - מספר המשתמשים, k - מספר הסרטים, m - מספר הקבוצות.

את כל הסרטים במערכת נשמור בעזרת 6 עצים שונים.
עץ אחד יכלול את כל הסרטים בשירות סטרימינג ממויינים לפי הId של הסרט. כלומר המפתח לכל איבר בעץ יהיה הID והערך יהיה טיפוס שמכיל את כל הפרטים של הסרט.
ניצור עץ עבור כל אחד מהז'נרים של הסרטים ועץ עבור כל הסרטים שכולם יהיו ממוינים לפי טיפוס הסרט באופן הבא: סידור בסדר יורד לפי דירוג ממוצע של סרטים, במקרה של שיוויון בדירוג אז בסדר יורד לפי צפיות ובמקרה של שוויון בצפיות אז בסדר עולה לפי ID. כלומר אם לסרט יש דירוג גבוה יותר מסרט אחר אז הסרט יהיה "מוקדם יותר" בעץ כלומר משמאלה בעץ. כל עץ של ז'נר מסוים יכיל אך ורק את הסרטים של הז'נר הזה.
מספר הצמתים בכל אחד מה6 עצים חסום על ידי מספר הסרטים בסטרימר סך הכל. כל סרט לוקח כמות קבועה של מקום אז לכן הסיבוכיות מקום הכוללת של ששת העצים היא $O(k)$

את כל המשתמשים במערכת נשמור בעזרת עץ אחד. המפתח של כל צומת יהיה הid של המשתמש והערך יהיה טיפוס שמכיל את כל הפרטים של המשתמש. כיוון שמספר הצמתים שווה למספר המשתמשים וכמות הזיכרון לצומת קבועה סיבוכיות המקום של העץ $O(n)$

את כל הקבוצות במערכת נשמור בעזרת עץ אחד. המפתח של כל צומת יהיה הid של הקבוצה והערך יהיה טיפוס שמכיל את כל הפרטים של הקבוצה. הטיפוס של כל קבוצה יכיל עץ של מצביעים לכל המשתמשים שבקבוצה. כמות הזיכרון שטיפוס שהעץ לוקח בלי.....**********

כל העצים ביחד מהווים כנדרש סיבוכיות מקום של $O(n+k+m)$

add_movie
נכניס טיפוס המכיל את כל הפרטים של הסרט לתוך עץ הסרטים הרגיל, עץ הסרטים של הז'נר של הסרט ועץ כל הסרטים עם המיון המיוחד. מספר הצמתים בכל אחד מהעצים האלה חסום על ידי מספר הסרטים בסטרימר k ולכן כל אחד מפעולות ההוספה האלה הן $O(\log k)$ וגם סך הכל ההכנסת היא $O(\log k)$

remove_movie
קודם נשיג את הפרטים של העץ מהעץ הרגיל ואז נשתמש בהם על מנת למצוצא ולהוציא את הצומת של הסטר מתוך שלושת העצים בו הוא נמצא (רגיל, ז'נר מדורג, כולם מדורג). כל אחת מה4 פעולות הנתונות חסומות על ידי $O(\log k)$ בדומה לסעיף הוקדם ולכן הסיבוכיות הכוללת היא $O(\log k)$

add_user, remove_user, add_group, remove_group
בכל אחד מהפעולות הללו נוציא או נכניס את האיבר המתאים לעץ המתאים לפי הid המתאים כהמפתח בעץ. מכיוון שהסיבוכיות של כל פעולה כזאת בעץ היא $O(\log x)$ כשx מספר האיברים בעץ אז הפעולות של המשתמשים יהיו $O(\log n)$ ושל הקבוצות יהיו $O(\log m)$

add_user_to_group
****
User_watch
נמצא את טיפוס המשתמש בעץ המשתמשים ואת הסרט בעץ הסרטים ואז נעדכן את השדות המתאימים שלהם. בנוסף נעדכן גם את השדות המתאימים במפתח של טיפוס הסרט בעצים הממויינים. נעשה זאת על ידי הוצאת הסרט מהעצים האלו והכנסה מחדש על מנת לעדכן את מיקומו בעץ.
כיוון שפעולות ההוצאה וההכנסה הן כמות חסומה של פעולות עם סיבוכיות $O(\log n),O(\log k)$ אז גם הסיבבויות הכוללת היא $\mathrm{O}(\log n + \log k)$
group_watch
****
get_all_movies_count
נחזיר את מספר הצמתים בעץ הסרטים הרגיל. כיוון שהגדרנו שהעץ שומר את מספר המתים בו כשדה הגישה אליו תיהיה O(1)
get_all_movies
נבצע סיור inorder בעץ הממוין לפי דירוג המתאים לgenre שהתקבל ונכניס את הid של הסרטים למערך לפי הסדר. נממש את הסיור inorder בדומה לאלגוריתם שהודגם בתרגול שבו סיבוכיות זמן היא $O(x)$ כשx מספר הצמתים בעץ וסיבוכיות המקום היא $O(1)$. כיוון שמספר הצמתים של העץ הממוין המתאים הוא k אם genre=none וk_{genre} במקרה אחר אז דרישות הסיבוכיות של הפונקציה מתקיימות.
get_num_view
נמצא את המשתמש המתאים מתוך עץ המשתמשים בסיבוכיות $\mathrm{O}(\log n)$ ואז פשוט נשיג מהטיפוס המשתמש את ערך הצפיות המתאים. ערך זה תמיד מעודכן כי כל פעם שהמשתמש צופה בסרט או שהקבוצה שהמשתמש חלק ממנה צופה בסרט אז טיפוס המשתמש מתעדכן בהתאם. סך הכל $\mathrm{O}(\log n)$ כנדרש.
rate_movie
נמצא את המשתמש ואת הסרט המתאים מעצי המשתמשים והסרטים בסיבוכיות $\mathrm{O}(\log n),O(\log k)$
נוודא שהמשתמש הוא vip אם נדרש ונעדכן את ממוצע הדירוגים של טיפוס הסרט. בנוסף נצטרך לעדכן באופן דומה גם את טיפוס הסרט ב2 ההופעות שלו בעצים אחרים ונעשה זאת על ידי פעולות הוצאה והכנס לעץ ככה שהסדר של הסרטים גם יתדעכן.
כיוון שפעולות ההוצאה וההכנסה הן כמות חסומה של פעולות עם סיבוכיות $O(\log n),O(\log k)$ אז גם הסיבבויות הכוללת היא $\mathrm{O}(\log n + \log k)$

get_group_recommendation
****