מבני נתונים 1
234218

תרגיל רטוב

1

מספר

הוגש ע"י:

יובל גושן	205810179
שם	מספר זהות
תובל גלוון	312419971
שם	מספר זהות

<u>ציון</u>:

לפני בונוס הדפסה:

כולל בונוס הדפסה:

נא להחזיר לתא מס':

מבני הנתונים

1. artist_by_id עץ AVL של אמנים, כאשר הkey הוא מספר האמן.
כל איבר בעץ מכיל מצביע למבנה נתונים של אמן המפורט בסעיף הבא.
2. Artist מבנה נתונים אמן מכיל את השדות:
   א. artist_id
   ב. num_of_songs
   ג. song_by_id מערך של שירים לפי id (במקום הi נמצא השיר הi) שכל איבר בו הוא מספר ההשמעות לאותו שיר
   ד. list_pointer מערך של מצביעים לאיבר ברשימה song_count_list שתפורט בסעיף 3 שבה אותו שיר נמצא
3. song_count_list מבנה נתונים של רשימה מקושרת דו כיוונית שכל איבר בה מכיל:
   א. streams ערך של מספר ההשמעות של כל השירים תחת אותה חולייה ברשימה
   ב. artist_count עץ Avl שמכיל מספרי אמנים (key = artist_id) שיש להם שירים בעלי streams השמעות
      1) כל עץ מכיל מצביע לעץ Avl נוסף: song_avl שמכיל את כל מספרי השירים (key הוא מספר השיר) שיש להם streams השמעות (זהה) לאותו אמן

הרשימה ממוינת לפי streams כך שהאיבר הראשון הוא הכי נמוך.
בנוסף נחזיק מצביע max_streams לאיבר האחרון ברשימה.

בכל עצי הAVL שנשתמש בהם – יהיו בכל Node המצביעים הבאים:

- left_son
- right_son
- father (מצביע לאבא)

בנוסף בכל עץ יהיה מצביע min לאיבר הקטן ביותר בו (השמאלי ביותר). עדכון מצביע זה לא משפיע על הסיבוכיות, כדי לעדכן אותו צריך רק:

- בהוספה לעץ – לבדוק אחרי ההוספה אם למינימום יש בן שמאלי, ואם כן להפוך את הבן השמאלי למינימום
- בהסרה מעץ – לבדוק אם האיבר שמוסר הוא המינימום, אם כן לעדכן את המינימום: אם יש לו בן ימני הוא הופך למינימום (לו לא יכולים להיות בנים כי זה עץ AVL), אם לא האבא הופך למינימום

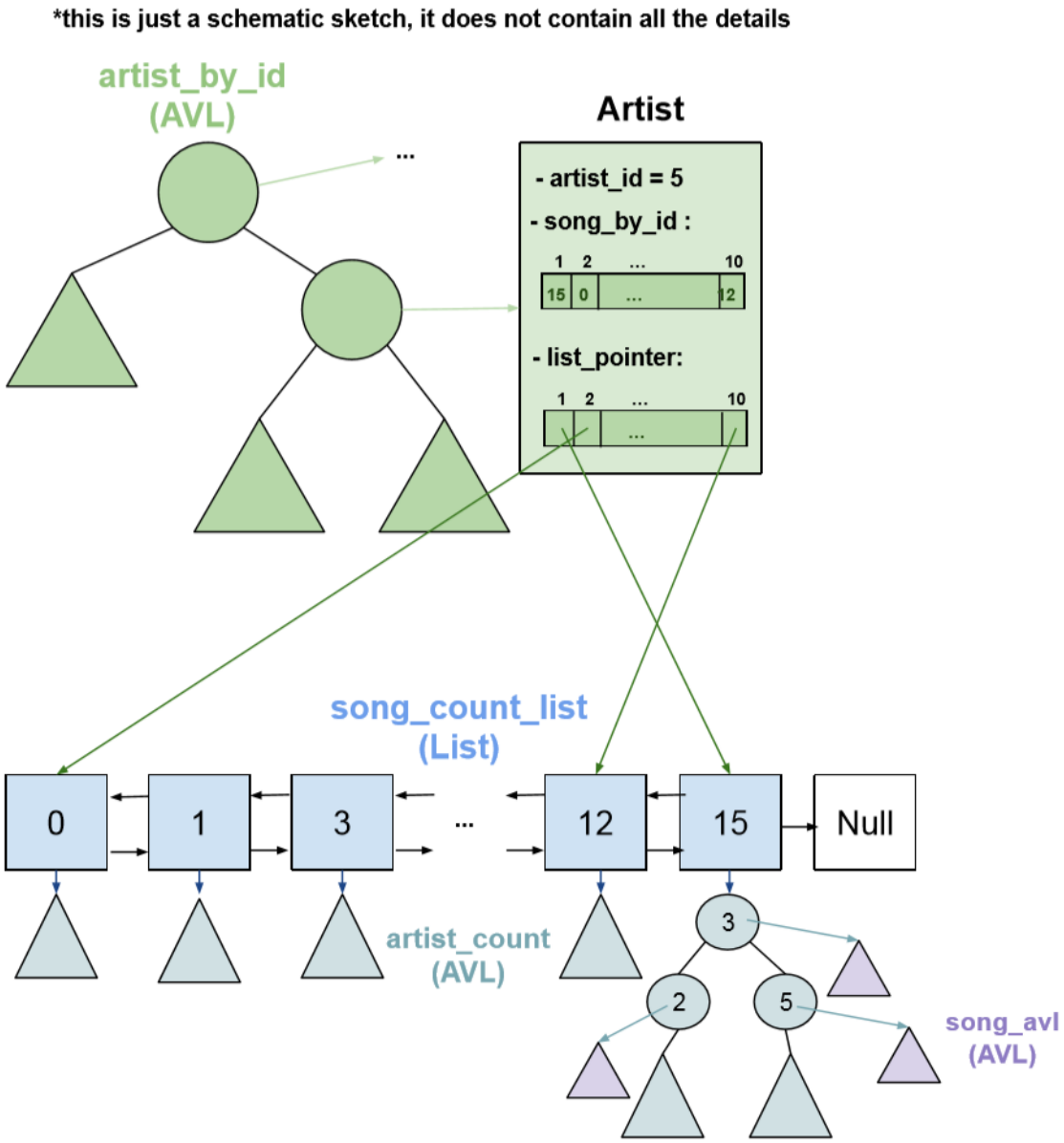
סיבוכיות מקום

נגדיר: n מספר אמנים, m מספר השירים הכולל

- artist_by_id מכיל איבר בעץ לכל אמן – O(n).
בנוסף כל איבר בעץ מכיל את כל השירים של אותו אמן וסה"כ העץ בתוכו מכיל מספר סופי של איברים עבור כל שיר במערכת (מספר השמעות, מצביע) – O(m)
- song_count_list במקרה הגרוע אצל כל שיר במערכת מושמע מספר שונה של פעמים, לכן יש m איברים ברשימה, ובכל איבר יש אמן אחד ושיר אחד תחתיו. באופן כללי, כל שיר יופיע בתוך הרשימה פעם אחת וכל אמן לכל היותר כמספר השירים שלו. סה"כ O(m)
קיימת גם אפשרות שלכל השירים אותו מספר השמעות ולכן הרשימה תכיל איבר אחד שבו עץ של כל האמנים ובכל אמן עץ של כל השירים שלו, סה"כ O(m+n)

קיבלנו סיבוכיות מקום :

$$O(m) + O(n) + O(m+n) = O(m+n)$$



פעולות

$init()$

1. מאתחלים עץ AVL ריק של אמנים (artist_by_id) – O(1)
2. מאתחלים רשימה של מספר השמעות – רק את האיבר של אפס השמעות והמצביע שלו לעץ artist_count ריק – O(1)

סיבוכיות זמן

כל האתחולים המפורטים הם ב(O(1.

$AddArtist(artist\_id, num\_of\_songs)$

1. ליצור מבנה נתונים Artist עבורו – O(1)
   א. לאתחל בו את מספר ההשמעות של כל השירים לאפס במערך songs_by_id – O(m)
   ב. נעדכן את המצביעים list_pointer כולם לחולייה אפס ברשימה song_count_list – O(m)
2. להכניס את המבנה מהסעיף הקודם לעץ AVL של האמנים artist_by_id – הכנסה רגילה לעץ Avl – O(logn)
3. נוסיף את האמן לעץ ברשימה song_count_list בחוליה הראשונה (של אפס השמעות) גודל העץ הוא לכל היותר מספר האמנים במערכת ולכן פעולה זו היא ((O(log(n, הכנסה לעץ AVL
4. בתוך האיבר שהוספנו בסעיף הקודם ניצור עץ AVL : song_avl ונכניס לתוכו את כל השירים של אותו אמן. מכיוון שאנחנו יודעים כבר באיזה סדר השירים מסודרים, אפשר להכניס אותם לעץ בסיבוכיות של O(m) ולא(O(mlogm כמו הכנסה רגילה של m איברים לעץ רגיל. אלגוריתם זה יוסבר בנספח : הכנסת איברים לעץ AVL כשיודעים את הסדר ב O(m) .

סיבוכיות זמן

פירטנו את הסיבוכיות של כל פעולה, סה"כ :

$$O(1) + O(m) + O(logn) + O(logn) + O(m) = O(logn + m)$$

$RemoveArtist(artist\_id)$

1. נמצא אותו בעץ artist_by_id -O(logn)
2. נשתמש במערך המצביעים list_pointer שלו כדי לגשת לכל איבר ברשימה בו שיר מסויים מופיע.
   א. עבור אותו שיר, נבדוק אם האמן שלו קיים בעץ artist_count (כי יכול להיות שכבר מחקנו אותו כשהיה שיר עם אותו מספר השמעות) – O(logn) (חיפוש בעץ AVL עם n אמנים לכל היותר)
   ב. אם הוא עוד לא נמחק :
      a. נמחק את העץ שירים שתחתיו song_avl – מחיקת עץ AVL בגודל m לכל היותר - O(m)
      b. נמחק את האומן מאותו עץ artist_count – O(logn) (מחיקה מעץ AVL עם n אמנים לכל היותר)

עבור כל השירים פעולה זו תקח במקרה הגרוע : כשלכל שיר של אותו אמן מספר השמעות שונה : O(mlogn) כי אנחנו מחפשים עבור כל שיר את האמן פעם אחת.

אם לכל השירים אותו מספר השמעות, אז כולם תחת אותו עץ AVL ואז זו מחיקה של m איברים מעץ אחד : O(m) ועוד חיפוש של אותו אמן אמן בעץ m פעמים, עדיין O(mlogn+m) = O(mlogn).

אם זה היה האיבר האחרון בעץ artist_count יש למחוק את אותו איבר מהרשימה

3. מוחקים את אותו אמן מartist_by_id ומשחררים את הזיכרון שלו – O(logn) גם פעולה זו היא מחיקה מעץ avl עם n אמנים

סיבוכיות זמן

$$O(logn) + O(m) + O(mlogn) + O(logn) = O(mlogn)$$

$AddToSongCount(artist\_id, song\_id)$

1. נמצא את האמן בעץ artist_by_id - O(logn)
2. נעלה במערך song_by_id את מספר ההשמעות לשיר ב1 – O(1)
3. דרך המצביע ב list_pointer ניגש לחוליה ברשימה song_count_list שמכילה שירים עם כמות השמעות זהה לשיר לפני ההוספה –O(1)
4. נבדוק אם החולייה הבאה ברשימה גדולה ב1
   א. אם לא – נוסיף לרשימה חולייה כזו שתהיה גדולה ב1 מהנוכחית – O(1)
   ב. אם כן ניגש לאותה חולייה – O(1)
   ג. נעדכן את list_pointer מArtist לחוליה החדשה – O(1)
   ד. נוסיף בעץ artist_count של החולייה (החדשה או הבאה) את אותו אמן ואליו נוסיף את השיר בעץ song_avl– O(logm + logn)
5. נחפש בעץ האמנים artist_count שבחולייה הישנה (יש מקסימום n איברים בעץ) את האמן – O(logn)
6. נסיר מהעץ song_avl (שבו לכל היותר m איברים עבור m שירים של אותו אמן) את השיר – O(logm)
7. אם העץ song_avl ריק נמחק את האומן מהעץ artist_count – O(logn)
8. אם העץ artist_count ריק נמחק את כל החולייה הישנה מהרשימה song_count_list – O(1)

סיבוכיות זמן

$$c_1O(logn) + c_2O(1) + c_3O(logm + logn) + c_4O(logm) = O(logn + logm)$$

$NumberOfStreams(artist\_id, song\_id)$

1. מוצאים את האמן בעץ artist_by_id – O(logn)
2. מחזירים את השיר במקום song_id במערך song_by_id של אותו אמן – O(1)

סיבוכיות זמן

$$O(logn) + O(1) = O(logn)$$

$GetRecommendedSongs(num\_of\_songs, *artists, *songs)$

1. ניגש לחוליה האחרונה ברשימה באמצעות max_streams – O(1)
2. ניגש לאיבר המינימלי ששמרנו אליו מצביע בעץ artist_count (מייצג את האמן עם הid הכי נמוך) של אותה החוליה – O(1)
3. ניגש באותו אמן בעץ השירים song_avl לאיבר המינימלי במצביע ששמרנו – O(1)
4. נכניס למערכים את הפרטים מתוך האיבר המינימלי ונמשיך לפי inorder עד שנסיים את העץ או שנגיע לm איברים שהוכנסו למערך. עבור כל איבר שמודפס בסדר inorder אנו מבצעים מספר סופי של פעולות (התחלנו מהאיבר המינימלי ולכן אין בהתחלה צורך לרדת בעץ ב(log(m) לכן עבור כל הדפסה - O(1)

(*) אנו יכולים לעלות מהאיבר המינימלי כיוון שאנו מחזיקים מצביע לאבא של כל איבר, ולא נעבור לאבא של איבר לפני שסיימנו להכניס את כל תת העץ שלו למערכים ולכן כך תמיד נשמור על סיבוכיות הזמן.

5. אם סיימנו את העץ ועוד לא הוכנסו m איברים להחזרה אז נעבור לעץ השירים של האומן הבא (inorder בעץ artist_count באותו אופן) ונכניס שירים למערך inorder עד שנגיע לסוף עץ האמנים או לm שירים – O(1) עבור כל שיר
6. אם סיימנו את כל האמנים באותה חולייה נעבור לחולייה הקודמת ברשימה song_count_list ונחזור לשלב 2 עד שהמערך יתמלא בm שירים - O(1) לעבור חולייה

סיבוכיות זמן

מכיוון שעבור כל שיר שאנחנו מחזירים אנחנו מבצעים O(1) פעולות, עבור m שירים נקבל סה״כ $O(m)$

(במקרה הגרוע הגענו לשורש ואחריו נרד את גובה העץ כדי להדפיס את הבא בגודלו אבל במקרה זה מספר השירים שהדפסנו הוא מסדר גודל של גובה העץ ולכן נשמור על הסיבוכיות.)

$Quit()$

1. מוחקים את העץ artist_by_id שכולל n אמנים – O(n) (שחרור כל אמן מהזכרון זה O(1))
2. עוברים כל חוליה ברשימה המקושרת song_count_list ומשחררים את כל העצים שתחתיה, סה״כ כל אמן יכול להופיע כמספר השירים שלו וכל שיר יכול להופיע פעם אחת, לכן סה״כ מוחקים O(m) איברים

סיבוכיות זמן

$$O(m) + O(n) = O(m+n)$$

נספחים

הכנסת איברים לעץ AVL כשיודעים את הסדר ב-O(M)

1. אם אין איברים נעצור
2. נקבע את האיבר האמצעי להיות שורש העץ
3. נחזור לפעולה 1 עבור האיברים המסודרים שמצד שמאל לאמצעי ונקבע אותם להיות תת העץ השמאלי (כולם קטנים מהשורש)
4. נחזור לפעולה 2 עבור האיברים המסודרים שמצד ימין לאמצעי ונקבע אותם להיות התת עץ הימני (כולם גדולים מהשורש)

סה״כ נקבל עץ שגובה כל תת עץ שלו מקיים את תנאי עץ avl כיוון שגובה כל תת עץ הוא חצי מהרשימה פחות האיבר האמצעי, ולכן ההפרש בין תתי העצים הוא לכל היותר 1.

סיבוכיות זמן

$$T(m) = T\left(\left\lceil\frac{m-1}{2}\right\rceil\right) + T\left(\left\lfloor\frac{m-1}{2}\right\rfloor\right) + 1 = 2\cdot T\left(\frac{m-1}{2}\right) + 1 =$$

$$= 2\cdot\left(2\cdot T\left(\frac{\left(\frac{m-1}{2}\right)-1}{2}\right)+1\right)+1 = 2\cdot\left(2\cdot T\left(\frac{m-2-1}{4}\right)+1\right)+1 =$$

$$= 2^2\cdot T\left(\frac{m-2^1-1}{2^2}\right)+2^1+1 = 2^2\cdot\left(2\cdot T\left(\frac{\left(\frac{m-2^1-1}{2^2}\right)-1}{2}\right)+1\right)+2^1+1 =$$

$$= 2^3 \cdot T\left(\frac{m - 2^2 - 2^1 - 1}{2^3}\right) + 2^2 + 2^1 + 1 = \cdots =$$

$$= 2^k \cdot T\left(\frac{m - 2^{k-1} - \cdots - 2^1 - 1}{2^k}\right) + 2^{k-1} + \cdots + 2^1 + 1 = \cdots =$$

$$=_{(*)} 2^{\log(m+1)} \cdot T(1) + 2^{\log(m+1)-1} + \cdots + 2^1 + 1 = (m+1) + \frac{m+1}{2} + \cdots + \frac{m+1}{2^k} + \cdots + 1 =$$

$$=_{(**)} (m+1) \cdot \left(\frac{1 - \left(\frac{1}{2}\right)^{\log(m+1)}}{1 - \frac{1}{2}}\right) = (m+1) \cdot \left(2 \cdot \left(\frac{m}{m+1}\right)\right) = 2m = O(m)$$

(*) $\frac{m-2^{k-1}-\cdots-2^1-1}{2^k} = 1 \rightarrow m = 2^k - 1 \rightarrow \ k = \log(m+1)$

(**) סכום סדרה הדנסית עם מנה $\frac{1}{2}$

קיבלנו עץ avl בסיבוכיות זמן - $O(m)$