מבנה נתונים

$\boldsymbol{mainSongRanking}$ – עץ דרגות AVL. כל מפתח בעץ מורכב משלושה איברים בעלי חשיבות יורדת מבחינת סידור האיברים בעץ: כמות השמעות, מספר אמן, מספר שיר. כלומר האיברים יסודרו תחילה לפי כמות השמעות בסדר יורד, אם יש התנגשות אז הם יסודרו לפי מספר אמן בסדר עולה, ואם קיימת התנגשות נוספת אז הם יסודרו לפי מספר השיר בסדר עולה. המידע בכל איבר בעץ זהה למפתח שלו.

$\boldsymbol{Artists}$ – טבלת ערבול דינאמית עם $chain\ hashing$ אליה יוכנסו איברים לפי מפתח של מספר אמן. נכניס את האיברים לפי שיטת הכפל עם קבוע $\bar{\phi}$.

כל איבר בטבלה יכיל את האיברים הבאים:

1. $songsByNum$ – עץ AVL בו המפתחות הם מספרי השירים והמידע הוא כמות ההשמעות.
2. $songsByRank$ - זהה למבנה $mainSongRanking$ כאשר בו יהיו שירים של אמן אחד בלבד, והמידע בתוך העץ יהיה מספר השיר המתאים למפתח.

כל עץ AVL מכיל בתוכו גם את הנתונים הבאים

1. $maxNode$ – האיבר עם המפתח הגדול ביותר
2. $amountOfNodes$ – כמות האיברים בעץ

<u>סיבוכיות מקום</u>:

$mainSongRanking$ הוא $O(n)$ כאשר n זו כמות השירים הכוללת במערכת.

סכום כל האיברים בתוך כל $songsByNum$ ובתוך כל $songsByRank$ בתוך $Artists$ הוא $O(n)$. שאר האיברים בתוך $Artists$ הם $O(m)$ כאשר m זו כמות האומנים במערכת.

סה"כ:

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

<u>הסבר על מימוש טבלת ערבול דינאמית עם $chain\ hashing$</u>:

עבור שימוש דינאמי בטבלת הערבול, כל פעם שכמות האומנים שווה לגודל הטבלה נגדיל את המערך פי 2 וניצור מערך חדש. את המערך החדש נמלא תחילה ב $NULL$, כלומר עוד $2m$ פעולות $O(m)$, ששומר על הסיבוכיות במקרה הגרוע עבור סדרת m פעולות $O(m)$.

באותו אופן, כל פעם שכמות האומנים קטנה ממחצית מגודל הטבלה, ניצור טבלה חדשה בגודל $m/2$ ונמלא אותה ב $NULL$, כלומר עוד $m/2$ פעולות $O(m)$, ששומר על הסיבוכיות במקרה הגרוע עבור סדרת m פעולות $O(m)$.

אחר כך נעבור על כל האיברים בטבלה המקורית ונעביר אותם לטבלה החדשה תוך עשיית ערבול מחדש.

בצורה כזו אנחנו עדיין שומרים על פעולת $O(1)$ משוערכת בממוצע על הקלט.

<u>הסבר על $maxNode$ בתוך עץ AVL</u>:

בכל הכנסה והוצאה של איבר שלוקחים $O(\log n)$ נבצע פעולה נוספת בסוף של מציאת האיבר עם ה $rank$ הגבוהה ביותר. כפי שראינו בכיתה פעולה זו לוקחת $O(\log n)$ ולכן הסיבוכיות של פעולת מחיקה והוספה של איבר לא משתנה.

מימושים

בכל המימושים במקרה של בעיה בהקצאת זיכרון נסיים את פעולת הפונקציה ונחזיר $ALLOCATION\_ERROR$.

$void * Init()$

<u>פעולות</u>

1. אתחול $mainSongRanking$ ריק
2. אתחול $Artists$ ריק בגודל 5 (הגודל הינו שרירותי כדי לקבל גודל התחלתי כלשהו)

<u>סיבוכיות זמן</u>

1. $O(1)$ – אתחול עץ AVL ריק, כפי שראינו בהרצאה
2. $O(1)$ –טבלת ערבול ריקה בגודל סופי 5 כאשר נמלא כל אחת מ 5 האיברים בה כ $NULL$.

סה"כ: $O(1)$ במקרה הגרוע

$StatusType\ AddArtist(void * DS, int\ artistID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים $artistID \leq 0 \ || \ DS == NULL$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ כבר קיים בתוך $Artists$
4. נוסיף את $artistID$ לתוך $Artists$ עם הנתונים
   a. $songsByNum$ ריק
   b. $songsByRank$ ריק

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ - בממוצע על הקלט המשוערך כפי שראינו בכיתה, חיפוש איבר בתוך טבלת ערבול
3. $O(1)$ – מספר קבוע של פעולות
4. $O(1)$ – כפי שראינו בהרצאה. אתחול $songsByNum$ וגם $songsByRank$ הוא אתחול עץ AVL ריק

סה"כ: $O(1)$ בממוצע על הקלט המשוערך.

$StatusType\ RemoveArtist(void * DS, int\ artistID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים $artistID \leq 0 \ || \ DS == NULL$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ לא קיים בתוך $Artists$ או אם $amountOfNodes > 0$ של $songsByNum$
4. נמחק את $artistID$ מתוך $Artists$ עם הנתונים

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ - בממוצע על הקלט המשוערך כפי שראינו בכיתה, חיפוש איבר בתוך טבלת ערבול
3. $O(1)$ – מספר קבוע של פעולות
4. $O(1)$ - בממוצע על הקלט המשוערך כפי שראינו בכיתה, מחיקת איבר מתוך טבלת ערבול. כאשר בתוך $songsByNum$ ובתוך $songsByRank$ אין איברים ולכן מחיקתם לוקחת $O(1)$.

סה"כ: $O(1)$ בממוצע על הקלט המשוערך.

$StatusType\ AddSong(\ void\ *\ DS, int\ artistID, int\ songID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים $artistID \leq 0\ ||\ DS == NULL\ ||\ songID \leq 0$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ לא קיים בתוך $Artists$
4. נחפש את $songID$ בתוך $songsByNum$ המתאים
5. נחזיר $FAILURE$ אם $songID$ כבר קיים בתוך $songsByNum$
6. נוסיף לתוך $songsByNum$ איבר חדש עם מפתח $songID$ ועם מספר השמעות 0
7. נוסיף לתוך $songsByRank$ איבר חדש עם מפתח
   a. כמות השמעות = 0
   b. מספר אמן = $artistID$
   c. מספר שיר = $songID$
8. נוסיף לתוך $mainSongRanking$ איבר חדש עם אותו מפתח

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ בממוצע על הקלט המשוערך – חיפוש בטבלת ערבול דינאמית (כפי שראינו בכיתה).
3. $O(1)$ – מספר קבוע של פעולות
4. $O(\log n)$ – חיפוש איבר בעץ AVL (כפי שראינו בכיתה).
5. $O(1)$ – מספר קבוע של פעולות
6. $O(\log n)$ – הוספת איבר לעץ AVL (כפי שראינו בכיתה).
7. $O(\log n)$ – הוספת איבר לעץ דרגות AVL (כפי שראינו בכיתה).
8. $(\log n)$ – הוספת איבר לעץ דרגות AVL (כפי שראינו בכיתה).

סה"כ: $O(\log n)$ בממוצע על הקלט המשוערך, כאשר n הוא מספר השירים הכולל במערכת.

$StatusType\ RemoveSong(\ void\ *\ DS, int\ artistID, int\ songID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים $artistID \leq 0\ ||\ DS == NULL\ ||\ songID \leq 0$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ לא קיים בתוך $Artists$
4. נחפש את $songID$ בתוך $songsByNum$ המתאים
5. נחזיר $FAILURE$ אם $songID$ לא קיים בתוך $songsByNum$
6. נחלץ מתוך האיבר המתקבל את מספר ההשמעות x
7. נמחק את האיבר $songID$ מתוך $songsByNum$
8. נמחק מתוך $songsByRank$ את האיבר עם מפתח
   a. כמות השמעות = x
   b. מספר אמן = $artistID$
   c. מספר שיר = $songID$
9. נמחק מתוך $mainSongRanking$ איבר עם אותו מפתח

<u>סיבוכיות זמן</u>

1. (1) – מספר קבוע של פעולות
2. $O(1)$ בממוצע על הקלט המשוערך – חיפוש בטבלת ערבול דינאמי (כפי שראינו בכיתה).
3. $O(1)$ – מספר קבוע של פעולות
4. $O(\log n)$ – חיפוש איבר בעץ AVL (כפי שראינו בכיתה).
5. $O(1)$ – מספר קבוע של פעולות

6. $O(1)$ – מספר קבוע של פעולות
7. $O(\log n)$ – מחיקת איבר מעץ AVL (כפי שראינו בכיתה).
8. $O(\log n)$ – מחיקת איבר מעץ דרגות AVL (כפי שראינו בכיתה).
9. $(\log n)$ – מחיקת איבר מעץ דרגות AVL (כפי שראינו בכיתה).

סה"כ: $O(\log n)$ בממוצע על הקלט המשוערך, כאשר n הוא מספר השירים הכולל במערכת.

$StatusType\ AddToSongCount(\ void\ *\ DS, int\ artistID, int\ songID, int\ count)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים
$artistID \leq 0\ ||\ DS == NULL\ ||\ songID \leq 0 || count \leq 0$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ לא קיים בתוך $Artists$
4. נחפש את $songID$ בתוך $songsByNum$ המתאים
5. נחזיר $FAILURE$ אם $songID$ לא קיים בתוך $songsByNum$
6. נחלץ מתוך האיבר המתקבל את מספר ההשמעות x
7. נמחק את האיבר $songID$ מתוך $songsByNum$
8. נכניס לתוך $songsByNum$ איבר חדש עם המפתח $songID$ ועם כמות השמעות $x + count$
9. נמחק מתוך $songsByRank$ את האיבר עם מפתח
   a. כמות השמעות = x
   b. מספר אמן = $artistID$
   c. מספר שיר = $songID$
10. נמחק מתוך $mainSongRanking$ איבר עם אותו מפתח
11. נכניס לתוך $songsByRank$ איבר חדש עם מפתח
   a. כמות השמעות = $x + count$
   b. מספר אמן = $artistID$
   c. מספר שיר = $songID$
12. נכניס לתוך $mainSongRanking$ איבר עם אותו מפתח

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ בממוצע על הקלט המשוערך – חיפוש בטבלת ערבול דינאמי (כפי שראינו בכיתה).
3. $O(1)$ – מספר קבוע של פעולות
4. $O(\log n)$ – חיפוש איבר בעץ AVL (כפי שראינו בכיתה).
5. $O(1)$ – מספר קבוע של פעולות
6. $O(1)$ – מספר קבוע של פעולות
7. $O(\log n)$ – מחיקת איבר מעץ AVL (כפי שראינו בכיתה).
8. $O(\log n)$ – הכנסת איבר לעץ AVL (כפי שראינו בכיתה).
9. $O(\log n)$ – מחיקת איבר מעץ דרגות AVL (כפי שראינו בכיתה).
10. $O(\log n)$ – מחיקת איבר מעץ דרגות AVL (כפי שראינו בכיתה).
11. $O(\log n)$ – הכנסת איבר לעץ דרגות AVL (כפי שראינו בכיתה).
12. $O(\log n)$ – הכנסת איבר לעץ דרגות AVL (כפי שראינו בכיתה).

סה"כ: $O(\log n)$ בממוצע על הקלט המשוערך, כאשר n הוא מספר השירים הכולל במערכת.

$StatusType\ GetArtistBestSong(\ void\ *\ DS, int\ artistID, int\ *\ songID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים $artistID \leq 0\ ||\ DS == NULL\ ||\ songID == NULL$
2. נחפש את $artistID$ בתוך $Artists$
3. נחזיר $FAILURE$ אם $artistsID$ לא קיים בתוך $Artists$ או אם $amountOfNodes\ == 0$ של $songsByNum$
4. נשמור בתוך $songID$ את $maxNode$ של $artistID$ שמצאנו מתוך $songsByRank$

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ בממוצע על הקלט המשוערך – חיפוש בטבלת ערבול דינאמית (כפי שראינו בכיתה).
3. $O(1)$ – מספר קבוע של פעולות
4. $O(1)$ – מספר קבוע של פעולות

סה"כ: $O(1)$ בממוצע על הקלט המשוערך

$StatusType\ GetRecommendedSongInPlace(\ void\ *\ DS, int\ rank, int\ *\ artistID, int\ *\ songID)$

<u>פעולות</u>

1. נחזיר $INVALID\_INPUT$ אם מתקיים
$artistID == NULL\ ||\ DS == NULL\ ||\ songID == NULL\ ||rank \leq 0$
2. נחזיר $FAILURE$ אם ה $rank$ גדול מ $amountOfNodes$ של $mainSongRanking$.
3. נחפש האיבר בתוך $mainSongRanking$ בעל ה $rank$ המתאים.
4. נשמור בתוך $artistID$ ובתוך $songID$ את המידע המתאים שיתקבל מהאיבר.

<u>סיבוכיות זמן</u>

1. $O(1)$ – מספר קבוע של פעולות
2. $O(1)$ – מספר קבוע של פעולות
3. $O(\log n)$ – חיפוש איבר בעל $rank$ נתון בתוך עץ AVL דרגות (כפי שראינו בכיתה)
4. $O(1)$ – מספר קבוע של פעולות

סה"כ: $O(\log n)$ במקרה הגרוע

$void\ Quit(\ void\ **\ DS)$

<u>פעולות</u>

1. מעבר על כל האיברים בתוך $Artists$ לפי הסדר, כאשר עבור כל איבר שהוא לא $NULL$
   a. מחיקת $songsByRank$
   b. מחיקת $songsByNum$
   c. מחיקת שאר הנתונים באיבר
2. מחיקת $mainSongRanking$

<u>סיבוכיות זמן</u>

סך כל האיברים בתוך כל $songsByRank$ ובתוך כל $songsByNum$ הוא יחדיו $2n$ כאשר מחיקת עץ AVL לוקחת $O(n)$. כמות האמנים היא m לכן בגלל ש $Artists$ הוא מסוג טבלת ערבול דינאמית נקבל כי גודל הטבלה הוא במקסימום $4m$. במקסימום כל התאים בטבלה ריקים מלבד אחד בו נמצאת רשימה מקושרת המכילה את כל האומנים, נעבור על תאים ריקים בסיבוכיות של $O(m)$ במקרה הגרוע. מעבר על כל האומנים לוקח גם הוא $O(m)$. לכן נוכל למחוק את $Artists$ ואת כל האומנים $O(m)$ והשירים $O(n)$ והמעבר על התאים הריקים $O(m)$ בסך הכל $O(m+n)$. מחיקת $mainSongRanking$ היא גם מחיקת עץ AVL שלוקחת $O(n)$.

סה"כ:

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