*צריך לבדוק-*

* *אלגוריתמי חיפוש בעצי ה-AVL ועמידה בסיבוכיות*
* *מערך שירים – מכיל פוינטרים או ~~מחלקה מסוג SONG~~?*
* *עץ AVL של מספר השמעות – מכיל פוינטר*

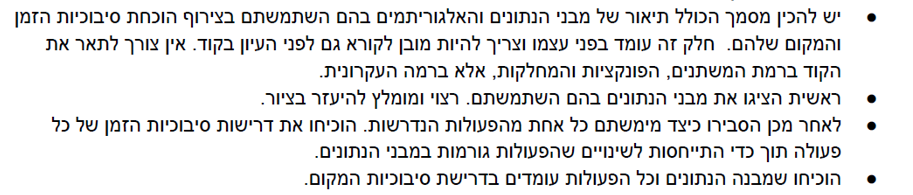
*מבנה הנתונים יורכב מהחלקים הבאים:*

*מחלקות כלליות*

1. *מחלקה גנרית של עץ AVL*
   1. *תומכת בכל הפעולות שצריך לעשות בעץ-*
      1. *הוספת צומת*
      2. *הוצאת צומת*
      3. *חיפוש צומת*
      4. *צריך לבנות נכון את האלגוריתמים?*
         1. *צריך לחשוב איך עושים את זה?*
   2. *כוללת גם NODE כללי גנרי*
      1. *שדות*
         1. *מצביע תבניתי ל-info (artist, song…)*
         2. *מצביע לבן ימני*
         3. *מצביע לבן שמאלי*
2. *מחלקה של אמן*
   1. *שדות:*
      1. *Int – Artist\_id? או שזה יהיה שם האמן?*
      2. *מערך – כלל השירים (מערך של מצביעים?)*
      3. *עץ AVL – מספר ההשמעות שיש לכלל השירים*
   2. *מתודות:*
      1. *Ctor*
         1. *בונה את השדה artist\_id דיפולטיבית*
         2. *בונה מערך בגודל N (מספר השירים בעת יצירת ה-Artist)*
            1. *הקצאה דינמית*
            2. *מכניס כפוינטר לרשימת ההשמעה 0*
         3. *בונה אובייקט מסוג AVL מסוג מספר השמעות?*
            1. *בהתחלה עם 0 צמתים (NULL)*
      2. *Dtor*
         1. *מוחק את השדה artist\_id דיפולטיבית*
         2. *מוחק את המערך*
            1. *מחיקה דינמית – delete*
            2. *צריך לדאוג למחיקה של הפוינטר שברשימה המקושרת המתאימה*
         3. *מוחק את האובייקט מסוג AVL*
            1. *מחיקה ע"י המתודה של AVL*
      3. *Cctor – למחוק (שלא תהיה אפשרות)*
      4. *Assign op – למחוק (שלא תהיה אפשרות)*
      5. *Getters*
         1. *Get\_song*
            1. *מחזיר SONG לפי אינדקס*
         2. *Get\_id*
            1. *מחזיר את ה-artistid*
      6. *Editors*
         1. *Add\_play\_to\_song*
            1. *מעלה ב-1 את מספר ההשמעות בשיר לפי אינדקס*
            2. *או*
            3. *מעביר את המצביע*
3. *מחלקה של שיר*
   1. *כנראה שמספיק להחזיק פוינטר – או שנעשה מחלקה (צריך לציין שדות ומתודות)*
4. *מחלקה של מספר השמעות?*
   1. *מספיר שה-key יהיה המספר, והאיבר עצמו יהיה עץ AVL של שירים (של INT המייצג SONGID)*

*מבנה הנתונים עצמו*

1. *X משתנים כלליים עבור מבנה הנתונים:*
   1. *שדכ*
2. *X – עץ AVL כללי של אמנים*
   1. *קיים אחד כזה באופן כללי*
   2. *עץ הבנוי מ-NODES מסוג artist*
3. *X – עצי AVL של מספר ההשמעות*
   1. *קיים אחד כזה עבור כל artist*
   2. *עץ הבנוי מ-NODES מסוג song*
4. *X – עצי AVL של שירים*
   1. *קיים אחד כזה עבור כל מספר השמעה*
   2. *כל צומת – SONGID*
5. *X – רשימה מקושרת של "מספר השמעות"*
   1. *נוצרת בעת האתחול הראשוני init*
   2. *קיימת אחת כזו*
   3. *מכילה איברים מסוג עץ AVL מסוג Artist*
      1. *(שיש להם את אותו מספר השמעות לפי האיבר)*
6. *דגע*

**

* *לא לשכוח טיפול בשגיאות*

***נתאר את מבני הנתונים בהם בחרנו להשתמש עבור התרגיל:***

*מחלקות משמעותיות שמימשנו:*

* ***AvlTree***
  + *עץ AVL גנרי אשר מאפשר את הפעולות כפי שלמדנו בכיתה – הוספה, הוצאה וחיפוש של איברים בתוכו. מימשנו את העץ כתבנית, כך שכל צומת ברשימה יכול להכיל איברים מכל טיפוס שהוא, כדי שנוכל ליצור עצים מסוגים שונים (עץ אמנים, עץ מספר השמעות ועץ שירים, וכדומה).*
    - *נציין כי מימשנו את העץ כך שבכל רגע נתון נחזיק את הצומת המינימלי בו (לכן נוכל להגיע אליה ב-).*
* ***StreamList***
  + *רשימה מקושרת דו-כיוונית אשר מאפשרת את פעולות ברירת המחדל של רשימה – הוספה, הוצאה וחיפוש של איברים בתוכה. רשימה זו למעשה אינה גנרית, אלא מתאימה ספציפית לדרישות התרגיל.*
  + *כל צומת ברשימה מייצג את מספר ההשמעות (הצומת הראשון – "0" השמעות, הצומת האחרון – מקס' ההשמעות לשיר במערכת ברגע נתון), וכן מכיל בתוכו איבר מטיפוס עץ של מצביעים לעץ השמעות.*
* ***MusicManager***
  + *מבנה כללי ל"מנהל המוזיקה", דרכו נאפשר את כל הפעולות הרצויות במערכת אשר הוגדרו בתרגיל. מבנה זה מכיל עץ אמנים (כלל האמנים אשר הוכנסו למערכת, להלן עץ האמנים) וכן רשימה מסוג StreamList (להלן רשימת ההשמעות).*
* ***Artist***
  + *להלן אמן. זהו מבנה המייצג כל אמן שהוכנס למערכת.*
  + *לאמן יש מזהה ייחודי (artist\_id, לפיו הוא גם יישמר בעץ האמנים), מערך של כלל שיריו (ובנוסף משתנה המכיל את מספר השירים הכולל), וכן הוא מכיל עץ של מספר השמעות.*

*מערך שירים:*

* + - *מערך בגודל מס' השירים הכולל של האמן (num\_of\_songs). כל איבר במערך יהיה מצביע לצומת ברשימה StreamList אשר מייצג מספר ההשמעות שהיו לשיר הנ"ל במערכת.*

*עץ של מספר השמעות:*

* + - *עץ של מספר השמעות יכיל את כל מספרי ההשמעות שיש לאמן עבור כל שיריו (למשל, אם לאמן 2 שירים עם 0 השמעות – בעץ זה יהיה צומת אחד המייצג "0" השמעות). כל צומת יסודר בעץ ע"פ מספר ההשמעות של השירים תחתיו.*
    - *בכל צומת של העץ הנ"ל, יהיה איבר של עץ שירים – עץ המייצג את כל השירים של האמן אשר הושמעו X פעמים (למשל, בדוגמה הקודמת, תחת הצומת של 05" השמעות שבעץ מספר ההשמעות – יהיו שני צמתים המייצגים את שני השירים של האמן, מסודרים לפי song\_id).*

*נתאר מקרה לדוגמה במערכת, אשר ילווה בציור:*

*נניח שבמערכת 5 אמנים;*

*לאמן 17 יש 4 שירים, והשיר עם מספר ההשמעות הגבוה ביותר הוא שיר 1, אשר הושמע 22 פעמים;*

*נניח גם ש-22 הוא מספר ההשמעות הגבוה ביותר במערכת עבור שיר מסוים.*

*במימוש שלנו, המערכת תיראה כך:*

* *קיים MusicManager, אשר למעשה מתאר את כלל המערכת שלנו (וכפי שציינו, מאפשר לבצע את כלל הפעולות שנדרשו בתרגיל).*
  + *בעץ האמנים שלו יש 5 צמתים, כאשר אחד מהם הוא האמן עבורו artist\_id = 17.*
    - *לאמן מס' 17 יש מערך שירים בגודל 4, וכן עץ השמעות עם מס' צמתים לפי כמות השירים שהושמעו מספר שונה של פעמים עבור האמן הנ"ל.*
      * *האינדקס 1 (אשר מייצג את שיר מס' 1) מכיל מצביע לצומת אשר מייצג "22" השמעות ברשימת ההשמעות.*
    - *תחת הצומת 22 נמצא עץ השירים של האמן הנ"ל אשר הושמעו 22 פעמים – במקרה של הדוגמה שלנו יהיה זה שיר מס' 1.*
  + *ברשימת ההשמעות שלו יש מספר מסוים של צמתים (לפי כמות השירים שהושמעו מספר שונה של פעמים בכלל המערכת), כאשר הצומת הראשון מייצג "0" השמעות, והצומת האחרון מייצג את מספר ההשמעות הגבוה ביותר במערכת עם שיר כלשהו – "22" השמעות במקרה של הדוגמה שלנו.*
    - *הצומת "22" מכיל עץ מסוג מצביע לעץ השמעות, ובין השאר מכיל את הצומת "17" אשר מייצגת את אמן מס 17, ומכילה מצביע לצומת "22" בעץ ההשמעות של אמן זה.*

***מימוש כל אחת מהפעולות הנדרשות***

*ראשית, נציג טענות עזר אשר תעזור לנו בהוכחות עבור העמידה בסיבוכיות הנדרשת:*

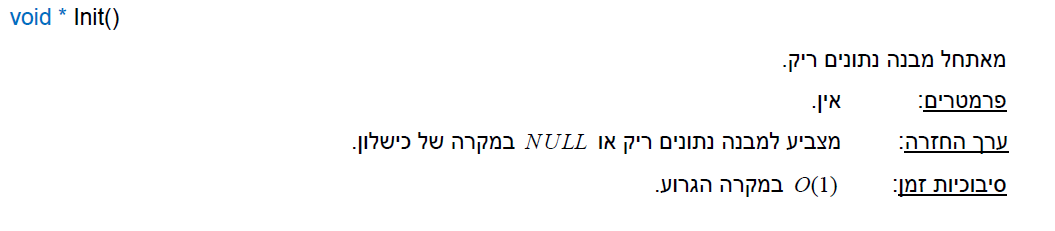
*טענת עזר שנתשמש בעת הוכחה על עמידה בסיבוכיות הנדרשת: נראה שבמידה ויש לך עץ ואתה מתחיל בצומת בעל האינדקס הכי קטן בעץ, ואתה עובר(מדפיס) לפי סדר r צמתים, אז תעשה סך הכל O(r) פעולות.*

*הוכחה: מכיוון שאנחנו מתחילים בצומת המינימאלי, הצומת הזו נמצאת בתחתית העץ (העלה הכי שמאלי) וכדי להגיע לצומת ה-r נעשה (מספר העליות שביצענו בעת המעברים)O, עתה מכיוון שאנחנו עוברים לפי סדר בכל עלייה כזו נצטרך או שלהדפיס את הצומת (במידה ועלית מבן שמאלי) או שלא (עלייה מבן ימני) ואז כבר הדפסת את הצומת הזו (כי הבן הימני יותר גדול מהצומת הנוכחי ואנחנו עוברים לפי סדר). לכן מספר העליות שסך הכל ביצענו הוא O(r), ולכן סך הפעולות שנעשה הוא O(r). כפי שרצינו להראות.*

***כמות השירים שהושמעו מספר שונה של פעמים בכלל המערכת (או עבור כל אמן) בהכרח קטנה או שווה למספר השירים הכולל במערכת (או עבור כל אמן):*** *נניח לאמן כלשהו יש X שירים. נניח בשלילה כי יש בעץ ההשמעות שלו X+1 צמתים ומעלה. מכיוון שלא ייתכן כי לשיר אחד יש 2 מספרי השמעות (שיר בהכרח הושמע מספר מדויק של פעמים בכל רגע נתון), נקבל כי בהכרח צומת מסוים בעץ ההשמעות יהיה ריק – וזה בלתי אפשרי, אחרת הצומת לא היה קיים (שכן אין לאמן שירים שהושמעו מספר כזה של פעמים). קיבלנו סתירה, ולכן הוכחנו את הטענה.*

***מספר האמנים הכולל במערכת בהכרח קטן או שווה למספר השירים במערכת:*** *המערכת לא מאפשר להכניס אליה שיר שאינו משויך לאמן, ולכן לכל אמן יש לכל הפחות שיר אחד.*

***Init:***

**

*בעת איתחול מבנה הנתונים שלנו, אנו יוצרים-*

* *עץ ריק של אמנים.*

*יצירת העץ קורית ב-, כיוון שאנו מאתחלים בסך הכל מספר קבוע של משתנים עבור המבנה של העץ.*

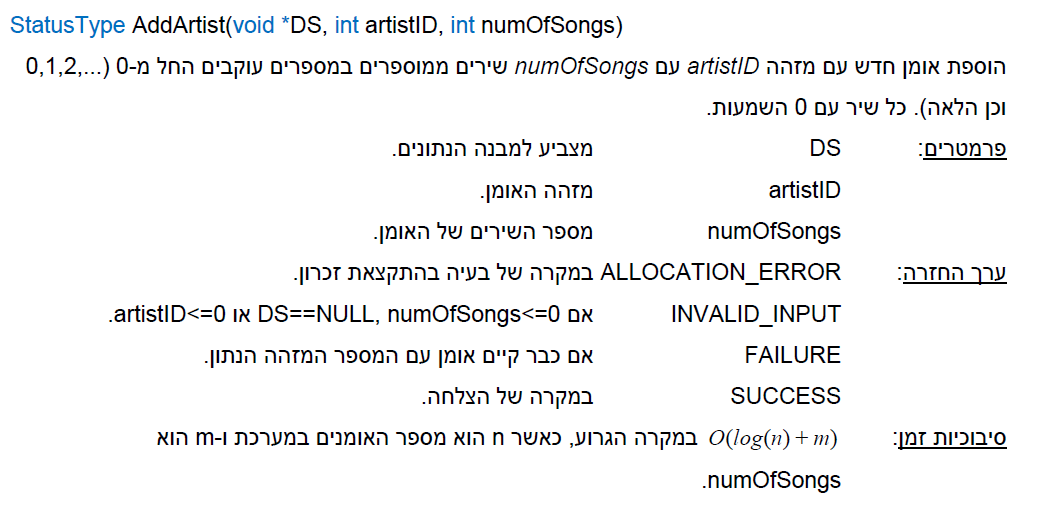
* *רשימה מקושרת של מספר ההשמעות, ובה יש צומת אחד המייצגת "0" השמעות. בצומת הנ"ל נוצר עץ ריק של מצביעים לאמנים.*

*גם יצירה זו קורית ב- (יצירת העץ, יצירת צומת אחד ויצירת רשימה אחת – כולם מאתחלים בסך הכל מספר קבוע של משתנים עבור המבנה).*

*מכיוון שאנחנו עושים לאורך כל תהליך האיתחול פעולות – אנו עונים על דרישות הסיבוכיות.*

*להתייחס לשגיאות?*

***AddArtist:***

**

*בעת הוספת אמן חדש למערכת, נשתמש בבנאי של Artist ליצור את האמן ולהוסיפו לעץ האמנים של MusicManager.*

*ביצירת האמן, אנו מאתחלים את המזהה שלו ואת מספר השירים הכולל ב-. בנוסף, אנו יוצרים מערך חדש בגודל m, ומאתחלים כל איבר בו להצביע לצומת "0" ברשימת ההשמעות – פעולה שלוקחת . לבסוף, אנו יוצרים עץ השמעות חדש עם צומת אחד של "0" השמעות (לוקח ), עבורו אנו גם יוצרים עץ שירים חדש שמכיל צמתים עבור כל השירים של האמן, ולכן יש לו m צמתים. את יצירת עץ השירים הנ"ל אנו עושים ב- – הסבר על אלגוריתם ההוספה של מערך ממויין – הכנסה רקורסיבית.*

*אזי, יצירת האמן לוקחת , כלומר לפי מה שלמדנו בכיתה.*

*בהוספת האמן שיצרנו ל-MusicManager, ייקח לנו להוסיף אותו במקום הנכון (כפי שלמדנו בכיתה על הוספת איבר לעץ AVL).*

*לכן, בסופו של דבר נבצע פעולות כנדרש (מטענה שראינו בכיתה).*

*להתייחס לשגיאות?*

*RemoveArtist*

**

*בעת מחיקת אמן קיים מהמערכת, אנו מוצאים תחילה את האמן בעץ האמנים של MusicManager ע"י חיפוש בעץ AVL, שכפי שראינו בכיתה לוקח .*

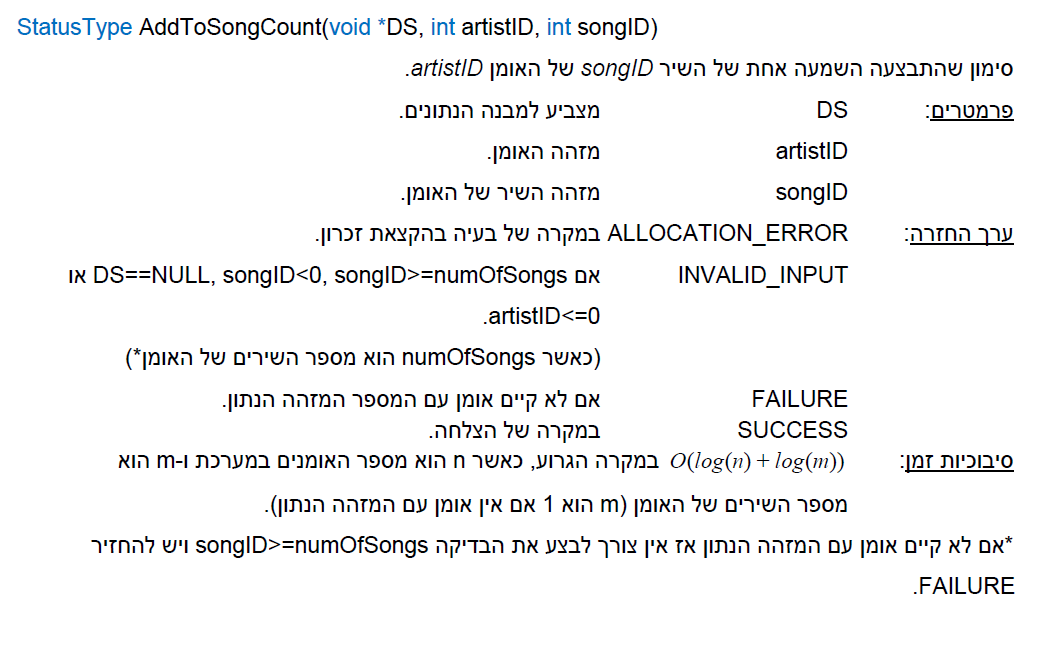
*משיש בידינו את האמן, אנו רצים בלולאה על מערך השירים הכולל שלו ובעזרתו מוחקים את הצמתים המצביעים על האמן מכלל העצים שבתוך כל צומת שברשימת ההשמעות:*

* *מתחילים במצביע שבאינדקס 0 במערך השירים הכולל, ומגיעים ב- לעץ המתאים תחת הצומת X ברשימת ההשמעות, כאשר X מתאר את מספר ההשמעות של השיר מס' 0.*
* *מוצאים את המצביע לאמן בתוך העץ הנ"ל ב-, כפי שתיארנו קודם עבור עץ AVL.*
* *נעזרים במצביע הנ"ל להגיע לעץ השירים שתחת עץ ההשמעות של האמן הספציפי בעץ האמנים של MusicManager, וע"י סיור in-order (שמתחיל בצומת הקטן ביותר, אליו אנו יכולים ב-) עוברים על כל השירים עם X השמעות ומשנים בתוך מערך השירים את ערך המצביע שבאינדקסים שמצאנו (לפי song\_id) ל-NULL (ראינו בכיתה כי סיור in-order שכזה ייקח , כאשר הוא מספר השירים של האמן עם X השמעות, כלומר מספר האיברים בעץ אשר בצומת X).*
* *כשמסיימים לעבור על כל עץ השירים הרלוונטי, עוברים לאינדקס 1 (האינדקס הבא במערך), ועושים את אותה פעולה. אם המצביע האינדקס הוא NULL, אנו עוברים ישר לאינדקס הבא.*
* *כך למעשה אנו רצים על כל האיברים במערך השירים, כלומר נעשה פעולות כדי לעבור על כלל המערך. בכך שאנו משנים את המצביעים ל-NULL במהלך הריצה בלולאה, אנו דואגים שבכל הריצות של הלולאה אנו עושים לכל היותר פעולות, כאשר הוא מספר הצמתים הכולל בעץ ההשמעות, ומטענה נקבל כי זהו מספר שחסום ב-m, ולכן אנו עושים לכל היותר פעולות נוספות.*

*בסופו של דבר, נבצע פעולות כנדרש (מטענה שראינו בכיתה).*

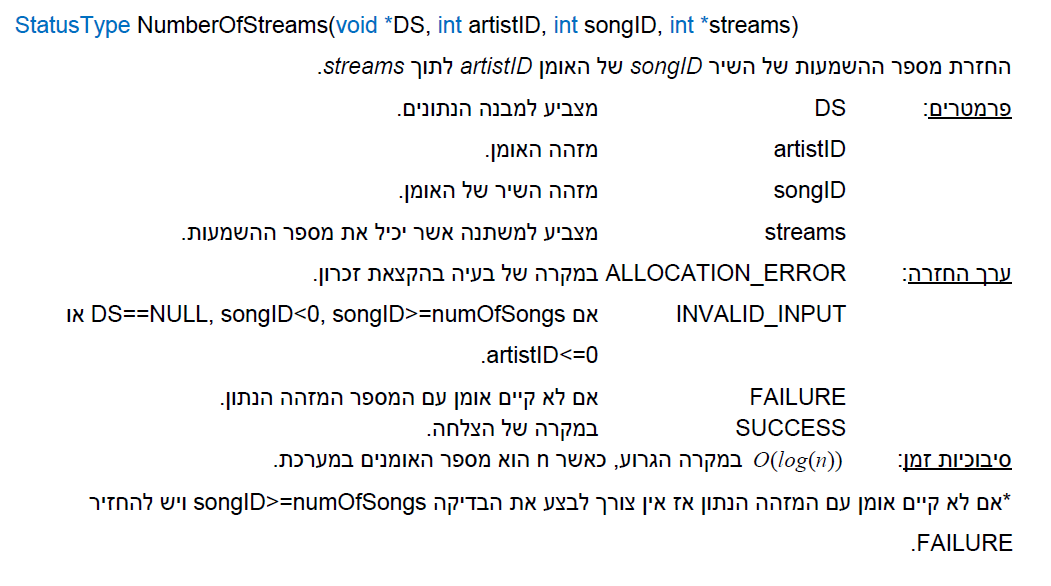
*להתייחס לשגיאות?*

*AddSongToCount*

**

*להתייחס לשגיאות?*

*NumberOfStreams*

**

*בעת החזרת מספר ההשמעות של השיר X של האמן Y, אנו מוצאים תחילה את האמן בעץ האמנים של MusicManager ע"י חיפוש בעץ AVL, שכפי שראינו בכיתה לוקח .*

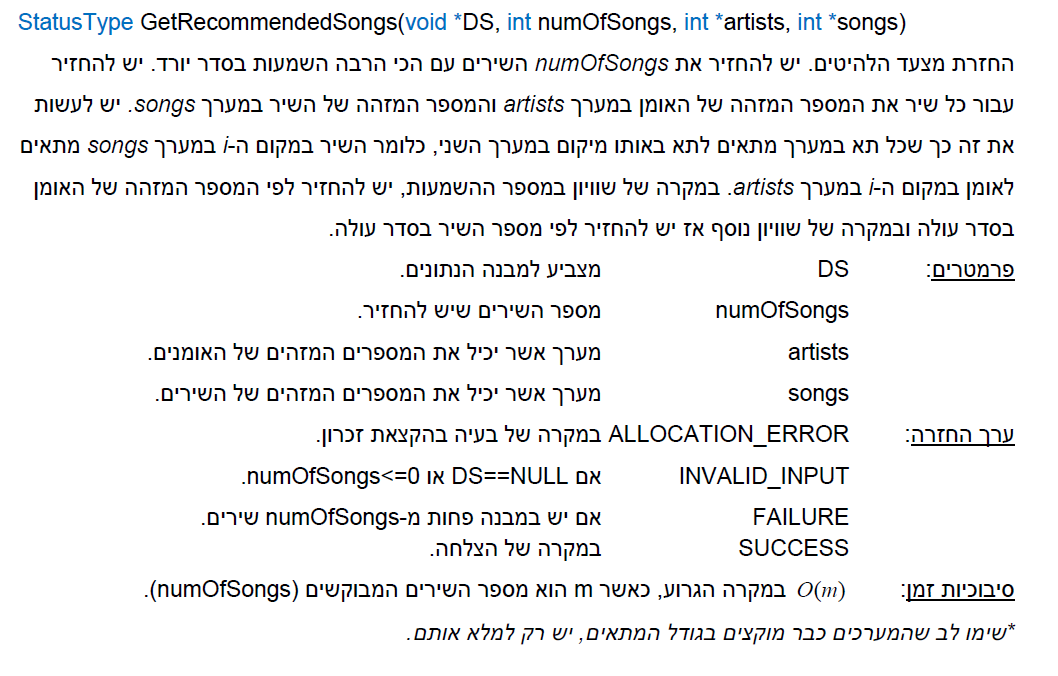
*כאשר יש בידינו את האמן Y, אנו יכולים לפנות לאינדקס X שבמערך השירים שלו ב- (המערך נמצא בתוך המחלקה של האמן, וכן פנייה לאינדקס ספציפי במערך לוקחת ). בתוך האינדקס הנ"ל ישנו מצביע לצומת ברשימת ההשמעות, צומת אשר מייצג את מספר ההשמעות של השיר X. הצומת הנ"ל מכיל בתוכו (במימוש שלנו) את מספר ההשמעות הרצוי, ולכן נבצע פעולות על מנת לקבל מספר זה.*

*מרגע שמספר ההשמעות בידינו, נבצע עוד פעולות על מנת לשמור את המספר בכתובת אליה מצביע , ונסיים.*

*לכן, בסופו של דבר נבצע פעולות כנדרש (מטענה שראינו בכיתה).*

*להתייחס לשגיאות?*

***GetRecommendedSongs:***

**

*בעת החזרת מצעד הלהיטים, אנו פונים תחילה לצומת האחרון ברשימת ההשמעות של MusicManager (אנו מחזיקים ברשימה מצביע לצומת האחרון שלה, לכן זה לוקח ), אשר מייצג את מספר ההשמעות הגבוה ביותר במערכת – נסמנו M.*

*הצומת של "M" השמעות מכיל עץ של מצביעים לצמתים "M" בעצי ההשמעות של כל האמנים להם יש לכל הפחות שיר אחד עם M השמעות. אנו ניגשים לצומת המינימלי בעץ המצביעים (נוכל לעשות זאת ב- שכן אנו מחזיקים מצביע לצומת המינימלי בעץ), שמחזיק למעשה מצביע לעץ השירים שהושמעו M פעמים של האמן עם ה-artist\_id המינימלי – נסמן אמן זה ב-.*

*בעץ השירים שהושמעו M פעמים של , אנו עוברים על השירים ע"י סיור in-order שייתן לנו את השירים בסדר עולה לפי song\_id (ושוב נוכל להתחיל מהאיבר המינימלי, כלומר השיר עם ה-song\_id המינימלי), כאשר עבור על שיר עליו נעבור – נכניס את הערך המתאים למערכי הקלט שקיבלנו כפרמטרים. אנו משתמשים ב-counter אשר מתחיל מ-0 וסופר את מספר השירים שכבר הכנסנו למערכים הנ"ל, וכך נוכל לדעת מה האינדקס הנכון במערכים להכניס אליו את הערך של השיר ושל האמן.*

*אם מספר השירים שעלינו להכניס למערכים (numOfSongs) קטן ממספר השירים שבעץ השירים הנ"ל, נעבור רק על חלק מהעץ, ואם גדול ממנו – נסיים לעבור על עץ השירים של האמן , נעבור לאמן הבא ונעשה את אותן הפעולות עבורו.*

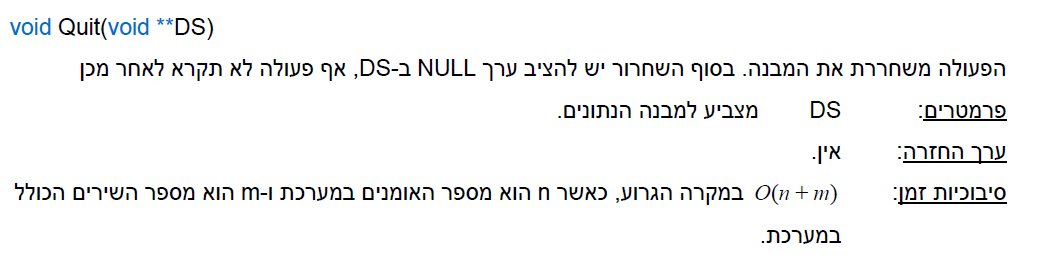
*כעת נתבונן במספר הצמתים שנעבור עליהם בעץ – נסמנו . מטענה נקבל כי סך הפעולות שאנו עושים הוא – והדבר נכון לגבי כל האמנים שנעבור עליהם. במקרה בו נעבור על כל האמנים שלהם יש לפחות שיר אחד עם M השמעות, נוכל לעבור לצומת הקודם לצומת "M" השמעות ב- (שכן כל צומת ברשימה מחזיק מצביע לצומת הקודם לו), ולהמשיך באותו סדר פעולות.*

*לכן, בסופו של דבר נבצע פעולות כנדרש:*

*(מטענות שראינו בכיתה).*

*להתייחס לשגיאות?*

***Quit:***

**

*תחילה ניזכר מטענה כי , ולכן (מטענה שראינו בכיתה).*

*בעת פעולת Quit, אנו למעשה מוחקים את MusicManager ומשנים את ערך המצביע DS ל-NULL (שינוי שקורה ב-). אם כן, סיבוכיות הפעולה תהיה כסיבוכיות המחיקה של MusicManager.*

*בעת המחיקה, אנו קוראים להורסים של עץ האמנים ושל רשימת ההשמעות שתחת MusicManager.*

* *סיבוכיות פעולת ההורס של עץ האמנים:*

*כל צומת בעץ מכילה אמן, ולכן מחיקת צומת תקרא להורס של האמן (ועוד פעולות על מנת למחוק את הצומת כולה). בעת מחיקת אמן, אנו רוצים למחוק את עץ ההשמעות שלו וכן את מערך השירים שלו:*

* + *מחיקת עץ ההשמעות של האמן תיקח O( פעולות, כאשר מסמל את מספר השירים שיש לאמן עם i השמעות.*
  + *מחיקת מערך השירים כולל של האמן לוקחת פעולות נוספות (כאשר מסמל את מספר השירים הכולל של האמן), מכיוון שהמערך הוא בסך הכל מערך של מצביעים בגודל (כאשר את האיברים שהמצביעים הצביעו אליהם נמחק בהמשך).*

*כלומר, נבצע פעולות למחיקת כל אמן:*

*(מטענות שראינו בכיתה).*

*בעץ הנ"ל יש n צמתים (כמספר האמנים), לכן עבור מחיקת כלל עץ האמנים נקבל:*

* *סיבוכיות פעולת ההורס של רשימת ההשמעות:*

*מחיקה של רשימת ההשמעות תבצע (במימוש שלנו) מחיקת צומת-צומת החל מהצומת האחרון, לצומת הלפני-אחרון וכך הלאה, עד שתמחק את כל הצמתים בה (וכן את עצמה ב-).*

*עבור כל צומת, נקרא להורס של הצומת, שלמעשה מוחק את עץ המצביעים שמכיל הצומת. פעולת המחיקה של כל עץ כזה תיקח פעולות, כאשר הוא מספר הצמתים בו (זו הסיבוכיות מכיוון שנבצע את המחיקה ע"י סיור in-order בו כל פעולת מחיקה היא למעשה מחיקת מצביע ב-).*

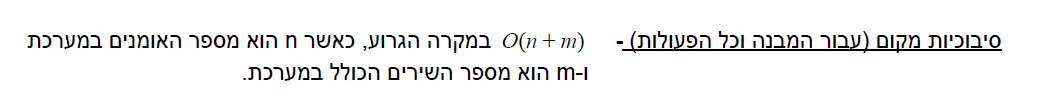
*נוכל לשים לב כי אם נסכום על כל ה--ים עבור כל הצמתים, נקבל מספר שחסום ע"י m – זאת מכיוון ש-, מס' הצמתים בעץ המצביעים שבצומת i, למעשה מתאר את מספר האמנים שיש להם שיר עם מספר השמעות כלשהו. מכאן, ש-.*

*אם נעבור על כלל הצמתים ברשימה, נקבל כי נבצע פעולות עבור כל צומת i שמחקנו, כלומר פעולות כנדרש (מטענה שראינו בכיתה).*

*לכן, בסופו של דבר נבצע פעולות כנדרש (מטענה שראינו בכיתה).*

*להתייחס לשגיאות?*

*סיבוכיות מקום*

**

*ב-quit מה שאנחנו עושים בעצם זה לקרוא לדסטרטור של ה-MusicManager. שהוא קורא לדיסטרקטור של של העץ אמנים שלנו ולדיסטרקטור של הרשימת השמעות. נראה עבור כל אחד שהוא עומד בתנאי הסיבוכיות.*

*עץ אמנים:*

*בעץ אמנים יש לנו עץ ובו n צמתים (כמספר האמנים), נעשה זום אין לצומת כזו, הצומת הזו מייצגת אומן מסויים (), באומן הזו יש עץ השמעות(num\_of\_streams\_tree) כאשר ה-key של כל צומת זה מספר ההשמעות וה-element של כל צומת זה עץ עם השירים שיש לאומן המסויים עם מספר השמעות של key.*

*נצטרך למחוק את העץ שירים הזה, המחיקה של עץ כזה היא O( כאשר זה מספר השירים שיש לאמן עם השמעות, בנוסף בסופו של דבר נצטרך למחוק את המערך full\_songs\_list ומכייון שזה בסופו של דבר פוינטר מחיקתו זה ב- (גודל המערך)O (מחיקת האיברים שהוא מצביע אליהם נעשית דרך המחיקה של הרשימה המקושרת שלנו). לכן סך הכל עבור מחיקה של אמן מעץ האמנים שלנו נעשה מס' פעולות שהוא*

*כאשר זה מספר השירים שיש לאמן a. זה עבור אמן ספציפי, עבור מחיקה של כל עץ האמנים נעשה באותה דרך:*

*.*

*רשימת השמעות:*

*נצטרך לעבור צומת צומת ובכל צומת למחוק את הצמתים בעץ הפוינטרים המתאים למספר ההשמעות של הצומת בו אנחנו נמצאים (את הדברים אליהם אנחנו מצביעים אנחנו מוחקים בעת מחיקת עץ ההשמעות שלנו). בכל צומת ברשימת ההשמעות נעשה מטענת העזר (מספר הצמתים בעץ הפונטרים) O ולכן סך הכל נעשה (סך הפויטרים)O וזה לכל היותר O(n+m).*

*נראה שהפונקציה אכן עונה על התנאי של הסיבוכיות שהוא O(m).*

*אנחנו מתחילים בצומת האחרון ברשימת ההשמעות שלנו (מכיוון שיש לנו פוינטר אליו ההגעה לצומת היא O(1)) שהיא מייצגת את מספר ההשמעות המקסימלי שיש לנו במערכת (נסמנו M).בצומת זו יש לנו עץ של פוינטרים לעצים המתאימים לM שירים לאומנים אשר יש להם לפחות שיר אחד עם M השמעות. אנחנו ניגשים לאיבר הראשון בעץ הפוינטרים (מכיוון שיש לנו פוינטר first תחת ה-AvlTree זה נעשה ב-O(1))- זה בעצם פויטר לעץ המתאים של האומן עם האינדקס הכי קטן שיש לו M השמעות (נסמנו ) . בעץ של השירים עם M השמעות תחת נעבור את השירים לפי הסדר (שוב מתחילים מהשיר עם האינדקס הכי קטן,הגעה אליו זה O(1) בגלל first ), כאשר על כל שיר שנעבור נשים את הערכים המתאימים למערכים, מכייון שיש לנו count שמתחיל מ-0 ושאותו אנחנו מגדילים ב-1 לאחר כל מעבר על שיר נדע תמיד את האינדקס המדוייק שבו אנחנו צריכים לשים את הערכים במערך. אנחנו נעבור על כל העץ (במקרה וnumSongs גדול ממספר השירים שיש ל- עם M השמעות ) או שנעבור רק על חלק מהעץ. נסמן את מספר הצמתים בעץ שנעבור עליהם ב-. מטענת העזר סך הפעולות שנעשה הוא O(). וזה נכון לכל העצים שלעיהם נעבור. מכיוון שאנחנו רצים בלולאות while עם count שמגדילים אותו ב-1 לאחר כל מעבר ומכיוון שמעבר מצומת לצומת הקודמת ברשימת השמעות היא O(1) על שיר נעשה סך הכל מספר פעולות של:*

*.*

*כנדרש, כאשר השוויון הראשון נובע מטענה שראינו בכיתה.*

*להלן איור הממחיש את מבנה המערכת*

*(האיור מכיל נתונים המתאימים עבור המקרה לדוגמה שתיארנו)*

***MusicManager***

*17*

*עץ אמנים*

*רשימת השמעות*

*0*

*22*

***Artist***

*artist\_id = 17*

*songs\_array*

*streams\_tree*

*total\_songs = 4*

*17*

*עץ מצביעים*

*1*

*2*

*3*

*0*

*מערך שירים*

*22*

*עץ השמעות*

*1*

*עץ שירים*