# מבני נתונים – יבש לתרגיל בית רטוב 1

מגישים: תום סמולין 313552739 הודיה ישראלי 315814939

תאריך הגשה: 25.05.2021

## תיאור מבני הנתונים:

להלן תיאור של "מרכיבי" המבנה שמימשנו.

השתמשנו בשני מבני נתונים עיקריים:

1. **עץ AVL** – אותו מימשנו בצורה גנרית, כפי שלמדנו בהרצאות. השתמשנו בו כדי למיין טיפוסי נתונים שונים. בנוסף לתכונות שראינו בכיתה שצריכות להתקיים בעץ, הוספנו לכל צומת גם מצביע ל"אב" שלה (כאשר השורש מצביע ל$nullptr$).
2. **מערך** – בו השתמשנו בצורה טריוויאלית כדי לשמור ולנהל בקלות את הציונים וכמות המכירות לכל דגם. פירוט נוסף בהמשך.

בנוסף, יצרנו את טיפוסי הנתונים הבאים:

1. **TypeInfo** – מכיל 3 שדות מטיפוס int ו2 מצביעים לטיפוסי int. העצמים מייצגים את: מס' הדגמים עבור סוג רכב מסוים, מספרו של הדגם הנמכר ביותר מסוג זה ומס' המכירות של אותו הדגם. 2 הפויינטרים, מצביעים למערכים אשר אורכם כאורך מספר הדגמים עבור אותו הסוג. לכל דגם "תא" בכל אחד מהמערכים. מערך ראשון, מייצג את הציון עבור כל רכב (בתא ה $i$ מופיע הציון של דגם $i$). המערך השני מייצג את מס' המכירות באותו האופן.
2. **ModelGradeInfo** – מכיל 3 שדות מטיפוס int עבור דגם מסוים במערכת. העצמים מייצגים את: ציון הדגם, סוג הדגם ומס' הדגם.
3. **ModelSaleInfo** – מכיל 3 שדות מטיפוס int עבור דגם מסוים במערכת. השדות מייצגים את: מספר מכירות הדגם, סוג הדגם ומס' הדגם.
4. **AvlTreeWrap** – מכיל עץ AVL ומצביע לצומת המינימלי בעץ. טיפוס זה מוכל בכל צומת ב*"עץ האפסים הגדול"* עליו יפורט בהמשך. העץ, מנהל וממיין עבור כל סוג את הדגמים בעלי ציון אפס – כלומר, אם יש דגם בעל ציון אפס, הוא מיוצג ומנוהל בעץ זה. כאמור, בנוסף לעץ ישנו מצביע לצומת המינימלית בעץ, אשר מאפשר התחלת סיור in-order בסיבוכיות הנדרשת בפונקציה GetWorst. פירוט בהמשך.

נשים לב כי גודל העצמים נקבע על פי גודל השדות הפנימיים שהם מחזיקים. הטיפוסים שיצרנו, מחזיקים עצמים מטיפוס int בגודל 4 בתים או כתובות בגודל 8 בתים (לעץ שמיישנו שדה פנימי יחיד – מצביע לשורש). לכל טיפוס כזה, מספר מצומצם של שדות פנימיים מ"הסוגים" שציינו, ולפיכך התייחסנו אליהם כאל עצמים פשוטים.

## מבנה הנתונים CarDealershipManager:

המבנה כולו מורכב מ-4 עצי AVL עיקריים:

1. **עץ סוגים** – ממוין לפי מספר הסוג. הסוג הקטן ביותר יופיע כצומת הכי שמאלית בעץ. לכל סוג רכב שקיים במערכת קיימת צומת בעץ זה. כל צומת מכילה TypeInfo.
2. **עץ הציונים** – ממוין לפי מפתח-וקטור של שלושה רכיבים: (מס' דגם, מס' סוג, ציון). לרכיב הציון בווקטור המשקל הרב ביותר והמיון נקבע קודם כל לפיו. במקרה של שוויון בציון , מיון הצמתים נקבע על פי רכיב מס' הסוג. במקרה של שוויון ברכיב זה, המיון יקבע על פי רכיב מס' הדגם. הצומת המייצגת את הרכב המסוים בעל הציון הנמוך ביותר, ממוקמת כצומת הכי שמאלית בעץ.
3. **עץ מכירות** - ממוין לפי מפתח-וקטור של שלושה רכיבים: (מס' דגם, מס' סוג, מס' מכירות). לרכיב מס' המכירות בווקטור המשקל הרב ביותר והמיון נקבע קודם כל לפיו. במקרה של שוויון ברכיב זה, מיון הצמתים נקבע על פי רכיב מס' הסוג. במקרה של שוויון ברכיב זה, המיון יקבע על פי רכיב מס' הדגם. הצומת המייצגת את הרכב המסוים בעל מס' המכירות הרב ביותר, ממוקמת כצומת הכי ימנית בעץ.
4. **עץ אפסים גדול** – ממוין לפי מספר הסוג. הסוג הקטן ביותר יופיע כצומת הכי שמאלית בעץ. רק לסוגים שלהם ברגע נתון קיימים דגמים בעלי ציון אפס, יש צומת בעץ זה. כל צומת מכילה AvlTreeWrap. כאמור, טיפוס זה מכיל עץ אפסים "קטן יותר" (מתייחס לscope של סוג הרכב) שממיין ועוקב אחר הדגמים בעלי ציון אפס של אותו הסוג. הדגם בעל המספר הקטן ביותר, יופיע כצומת הכי שמאלית בעץ זה. צומת בעץ האפסים הגדול מכילה בנוסף מצביע לצומת המינימלית בעץ האפסים הקטן יותר – כדי לאפשר סיור inorder שעומד בסיבוכיות הנדרשת ב GetWorst.

כאמור, עץ האפסים הגדול מכיל צמתים השייכים לסוגים השונים, כך שבכל צומת מופיע עץ שנכנה אותו – ***עץ אפסים קטן.***

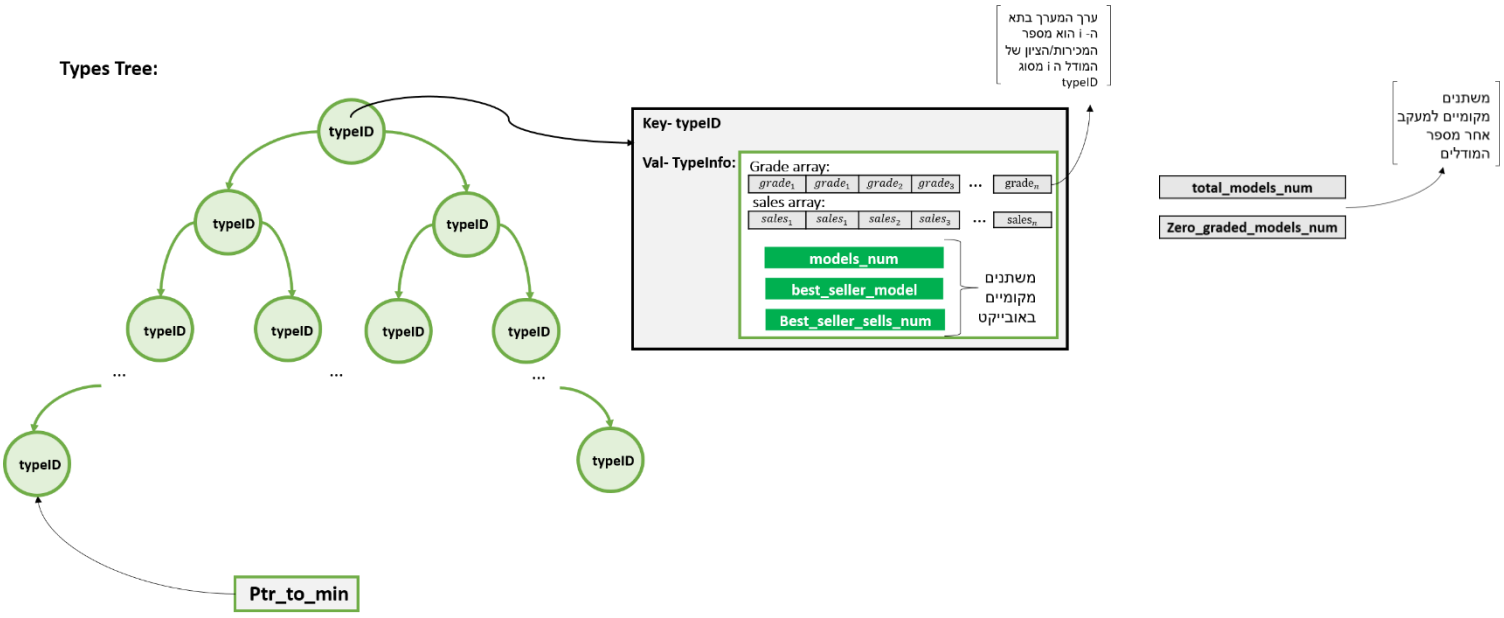
4.1. **עץ אפסים קטן** – מכיל צמתים מהסוג שמופיע בעץ הניקוד. רק שבמקרה זה, במפתח הווקטור ערכי הציון והסוג שווים, כאשר הציון הוא 0 (שכן עץ האפסים הקטן מכיל רק דגמים של אותו הסוג בעלי ציון אפס). המיון נקבע ע"פ מס' הדגם. לכן הצומת הקטן ביותר בעץ (שיופיע כצומת הכי שמאלית בעץ) ייצג את הרכב בעל מס' הדגם הקטן ביותר מאותו הסוג, בעל ציון אפס.

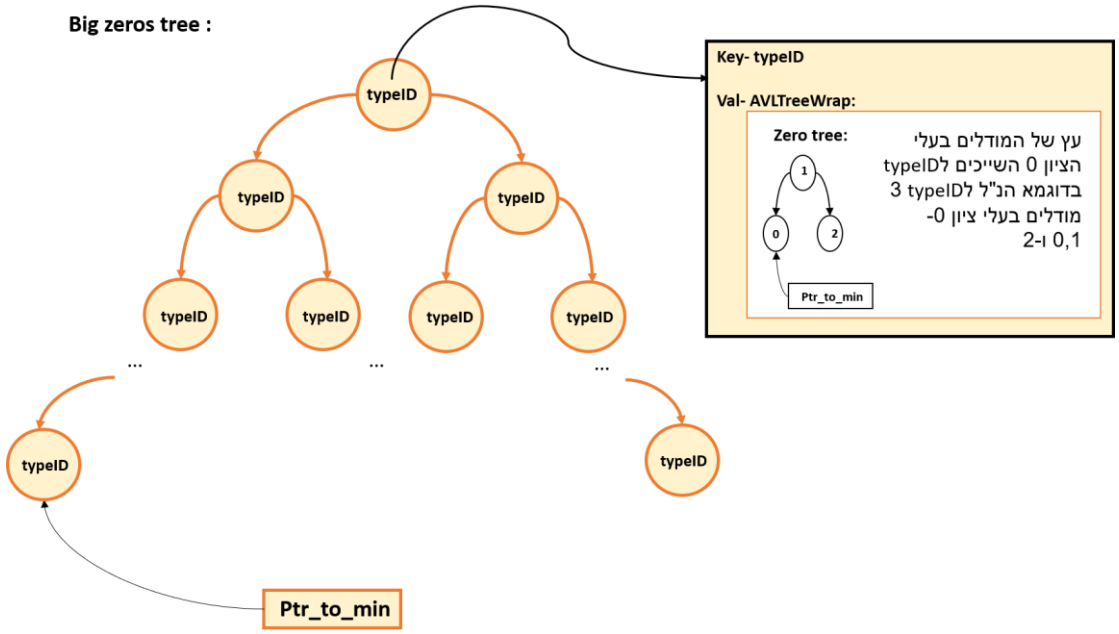
**בנוסף,** מבנה הנתונים הנ"ל מחזיק 3 מצביעים נוספים (כדי לעמוד בדרישות הסיבוכיות בהמשך) ל:

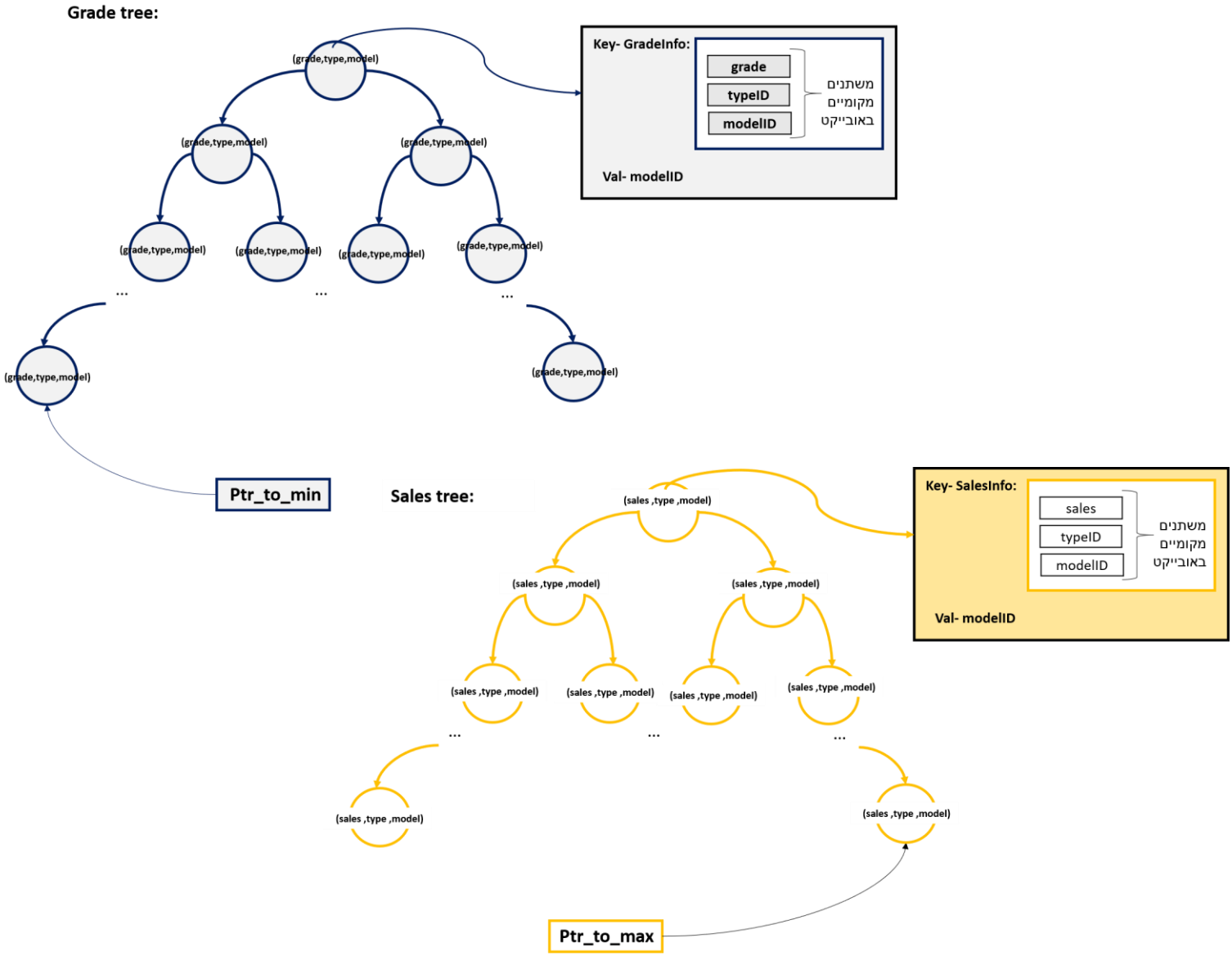
- צומת המינימלית בעץ האפסים הגדול
- צומת המינימלית בעץ הציונים
- צומת המקסימלית בעץ המכירות

וכמו כן שני משתנים מקומיים המונים את מס' הרכבים הכולל במבנה, ואת מס' הרכבים בעלי ציון אפס ברגע נתון.

**נתאר את מבנה הנתונים באמצעות סרטוט:**






**פעולות עיקריות בעץ והוכחת סיבוכיות:**

- **אתחול עץ ריק:** $\boldsymbol{O(1)}$
בעת אתחול העץ כל שעושים הוא לאתחל מצביע מסוג החולייה לשורש ולאתחלה ב- $nullptr$.
- **חיפוש:** $\boldsymbol{O(log\ n)}$ **(באשר $\boldsymbol{n}$ הוא גודל העץ).**
מעצם המבנה של עץ AVL מאחר וגובה העץ הוא לוגריתם של גודל העץ. (כפי שלמדנו בהרצאה תוחלת גובה העץ שייכת לקבוצת $O(\log n)$).
- **הוספה איבר:** $\boldsymbol{(log\ n)}$ **(באשר $\boldsymbol{n}$ הוא גודל העץ).**
כפי שנלמד בהרצאה מחיר הפעולות שמתבצעות בהכנסת איבר הוא להלן:
מציאת המקום הדרוש להכנסה- $O(\log n)$.
הוספת החולייה למקום זה- $O(1)$ (דורש עדכון מצבעים בלבד).
מציאת המקום בו מופר האיזון- $O(\log n)$.
תיקון האיזון- $O(1)$
- **הוצאת איבר:** $\boldsymbol{(log\ n)}$ **(באשר $\boldsymbol{n}$ הוא גודל העץ).**
מציאת האיבר והוצאתו- $O(\log n)$.
מציאת המקום בו מופר האיזון- $O(\log n)$.
תיקון האיזון, מאחר ובמקרה של הוצאה ייתכנו מספר גלגולים (לכל היותר גלגול אחד לרמה)- $O(\log n)$.
- **הריסת עץ-** $\boldsymbol{O(n)}$ **(באשר $\boldsymbol{n}$ הוא גודל העץ).**
הריסת העץ נעשה בסיור post order על איברי העץ ומחיקת כל איבר.

הפעולות הנדרשות:

❖ הערה: בכל הפעולות הבאות, מתבצעת בדיקה על תקינות הפרמטרים ע"י מספר קבוע של פעולות (בדיקות שלא נעשות ע"י מספר קבוע של פעולות מפורטות בתיאור הפעולות).

1 $\boldsymbol{void} * Init(\ )$

אתחול של מבנה הנתונים, ע"י יצירה של עצם מסוג CarDealershipManager. בזמן היצירה מאותחלים: ארבעת העצים העיקריים (השמה של $nullptr$ לשורש של כל עץ), שלושת המצביעים (השמה של $nullptr$) ושני המשתנים מטיפוס $int$.

<u>סיבוכיות זמן: $O(1)$</u>

השמות אלו מבוצעות בסיבוכיות זמן של $O(1)$ כפי שנלמד בהרצאה (מס' קבוע של פעולות בסיבוכיות של $O(1)$).

2 $\boldsymbol{StatusType}\ AddCarType(\boldsymbol{void} * DS,\ \boldsymbol{int}\ typeID, \boldsymbol{int}\ numOfModels)$

הוספת סוג רכב חדש למערכת. בפעולה זו אנו **ראשית** מוסיפים צומת של הסוג החדש לעץ הסוגים. לאחר ייצור הצומת הנ"ל, אנחנו מאתחלים את שני המערכים בTypeInfo – אשר מוכל בצומת זו. כל דגם מתחיל עם ציון אפס ומס' מכירות השווה לאפס – לכן שני המערכים מאותחלים להחזיק ערך 0 בכל תא.
**שנית**, כיוון שהדגמים של הסוג שנוספו בעלי ציון אפס, אנו מוסיפים צומת של הסוג החדש *לעץ האפסים הגדול* ומייצרים עבור צומת זו *עץ אפסים קטן* שצמתיו מייצגים את הדגמים של הסוג.
לאחר הכנסות אלו, אנו מעדכנים את המצביעים הרלוונטיים לכל הצמתים המינימליים – הן עבור עץ האפסים הגדול והן עבור עץ האפסים הקטן בצומת של עץ האפסים הגדול.

<u>סיבוכיות זמן: $O(\log(n) + m)$- כאשר $m$ הוא מספר הדגמים של הסוג שנוסף ו $n$ הוא מספר הסוגים במערכת.</u>
הכנסת צומת לעץ הסוגים ולעץ האפסים הגדול נעשית בסיבוכיות של $\log(n)$ עבור כל עץ לפי הנלמד בהרצאות (במקרה הגרוע ישנם $n$ סוגים במערכת וקיימים עבורם צמתים הן בעץ הסוגים והן בעץ האפסים הגדול. העץ הינו עץ *AVL* ולכן הסיבוכיות הנ"ל).
מס' הדגמים של הסוג החדש הוא $m$. לכן אתחול שני המערכים נעשה בסיבוכיות של $O(m)$.
יצירת עץ *AVL* מרשימה ממוינת של דגמים (כאמור, בעץ האפסים המיון הוא לפי מס' הדגם), נעשית בסיבוכיות של $O(m)$ כיוון שבכל שלב, בוחרים את המספר האמצעי מהטווח הנתון להיות השורש של תת העץ החדש שמייצרים, וממשיכים בצורה רקורסיבית לבנות את תתי העצים שלו, עם האיברים הגדולים ממנו בתת העץ הימני והאיברים הקטנים ממנו בתת העץ השמאלי.
לכן הנוסחא המתקבלת היא:

$$z(m) = 2T\left(\frac{m}{2}\right) + C = 2^{\log(m)} \cdot T(1) + C_{new} = m + C_{new} \rightarrow T(m) = O(m)$$

עדכון המצביעים לצמתים המינימליים נעשה בסיבוכיות $\log(n)$ עבור עץ האפסים הגדול ו $\log(m)$ עבור עץ האפסים הקטן – שכן מדובר במסלול שעובר מהשורש וממשיך רק על הבנים השמאליים, עד שאין עוד לאן להתקדם (הסיבוכיות שצוינה מתארת את מספר השלבים שיש לעבור במסלול, שכן זהו בקירוב גובהו המקס' של העץ כפי שנלמד, בהתאם למס' הצמתים).
סה"כ נקבל שסיבוכיות הזמן מסתכמת ב: $\boldsymbol{O(\log(n) + m)}$ כנדרש.

3 $\boldsymbol{StatusType}\ RemoveCarType(\boldsymbol{void} * DS,\ \boldsymbol{int}\ typeID)$

מחיקת סוג רכב ממבנה הנתונים . תחילה מוצאים את הסוג הנתון *בעץ הסוגים* (ומוודאים שהוא קיים) *ובעץ האפסים הגדול*. בצומת המתאימה *בעץ האפסים הגדול*, מוחקים את *עץ האפסים הקטן*. לאחר שהסרנו את *עץ האפסים הקטן*, אנו מתפנים להסיר את הצומת של הסוג הנ"ל *מעץ האפסים הגדול*.
לאחר שטיפלנו בעצי האפסים ניגש לטפל בדגמים שלהם ציון שונה מ-0. כיוון שמצאנו את הצומת של הסוג *בעץ הסוגים*, יש לנו גישה למערכים של הציונים ושל מספרי המכירות של הדגמים. נעבור בלולאה על כל הדגמים של הסוג. בכל איטרציה, נתבונן בציון הנתון ובמס' המכירות של הדגם באיטרציה. אם הציון שווה לאפס, הרי שטיפלנו בחוליה שמייצגת את הציון של הרכב. אחרת, בידינו המפתח (יש לנו את מס' הדגם, מס' הסוג ואת הציון שלו) ולכן נמצא את החולייה המתאימה של הדגם *בעץ הציונים* ונסיר אותו מהעץ. באופן זהה נטפל במכירות – באותה איטרציה נוכל להרכיב את המפתח *לעץ המכירות*. אם מס' המכירות של הדגם הוא אפס, אזי לא הוספנו אותו *לעץ המכירות* מלכתחילה. אחרת, נאתר אותו *בעץ המכירות* ונסיר אותו ממנו.

לאחר שטיפלנו בכל הדגמים השייכים לסוג בעץ הציונים, עץ האפסים הגדול ובעץ המכירות – נסיר את הצומת של הסוג מעץ הסוגים. בשלב זו לא נותרו ייצוגים של הסוג או של דגמיו במערכת. מעדכנים את המצביעים לצמתים המינימליים/מקסימליים *בעץ הציונים, עץ האפסים הגדול ועץ המכירות*.

<u>סיבוכיות זמן: $O(\log(n) + m \cdot \log(M))$- כאשר $M$ הוא מס' המודלים $m$ מס' הדגמים של הסוג שהסרנו $n$ מס הסוגים המערכת.</u>

איתור הצומת בעץ הסוגים ובעץ האפסים הגדול נעשית ב $O(\log(n))$ לכל עץ. מחיקת עץ האפסים הקטן נעשית בסיבוכיות של $O(m)$ שכן העץ נמחק בסיור *post-order* (ולכל היותר $m$ דגמים בעץ הקטן). הסרת צומת הסוג *מעץ האפסים הגדול*, נעשית בסיבוכיות $O(\log(n))$ כפי שנלמד.

לאחר מכן, אנו עוברים על כל דגם של הסוג – ומסירים אותו במידת הצורך *מעץ הציונים* ו/או *מעץ המכירות*. כל הוצאה כזאת נעשית ב$O(\log(M))$ כיוון שבעצים אלו יכולים להופיע כל הדגמים במערכת. כיוון שאנו עוברים על כל הדגמים של הסוג, סה"כ הסיבוכיות של הסרת החוליות הנ"ל מעצי הציונים והמכירות היא $O(m \cdot \log(M))$ (מונים הסרה של כל דגם, ומקבלים את הסיבוכיות הנ"ל עד כדי קבוע (2)).

הסרת צומת הסוג מעץ הסוגים נעשית גם היא בסיבוכיות של $O(\log(n))$ . עדכון המצביעים לצמתים הקיצוניות נעשה בסיבוכיות של $O(\log(M))$ עבור עצי הציונים והמכירות ו $O(\log(n))$ עבור עץ האפסים הגדול, כפי שהסברנו בפעולה הקודמת.

נשים לב כי $m = O(m \cdot \log(M))$.

סה"כ נקבל שסיבוכיות הזמן מסתכמת ב: $\boldsymbol{O(\log(n) + m \cdot \log(M))}$.

4 $\boldsymbol{StatusType}\ SellCar(\boldsymbol{void} * DS,\ \boldsymbol{int}\ typeID, \boldsymbol{int}\ modelID)$

מכירת דגם מכונית של סוג רכב מסוים. תחילה מוצאים את הסוג הנתון *בעץ הסוגים* (ומוודאים שהוא קיים). כעת ניגש למערכי הציונים והמכירות של הסוג במקום המתאים, כדי שנוכל לעדכן את הערכים עבור הדגם המבוקש. נעדכן את הדגם הנמכר ביותר עבור הסוג, במידה ויש צורך בעדכון זה.

**אם** הציון של הדגם טרם העדכון היה אפס, ניגש לצומת הסוג בעץ האפסים הגדול, ונסיר את הצומת המתאימה לדגם מעץ האפסים הקטן. נשים לב שבמידה והתרוקן עץ האפסים הקטן, לפי איך שהגדרנו את מבנה הנתונים, נמחק את צומת הסוג מעץ האפסים הגדול.

לאחר הטיפול הנוגע לעצי האפסים, נכניס צומת חדשה ומעודכנת של הדגם *לעץ הציונים*.

**אחרת** (אם הציון הישן שונה מאפס), נסיר את הצומת המתאימה לדגם *מעץ הציונים*, נעדכן את המפתח ונכניס את הצומת מחדש *לעץ הציונים*$(*)$.

**כעת נשים לב** כי ייתכנו מקרים של דגמים בעלי ציון שלילי במערכת. במקרה שנמכר רכב בעל ציון שלילי והציון החדש שווה לאפס – נסיר את הצומת המתאימה *מעץ הציונים*, ונכניס צומת חדשה *לעץ האפסים הקטן* המתאים (ברוך במציאת צומת הסוג המתאים בעץ האפסים הגדולים, או בהכנסת צומת זו *לעץ האפסים הגדול* במידה והוא לא קיים בו). אם הציון משתנה לציון השונה מאפס, נפעל כמו ב$(*)$.

**לאחר שטיפלנו** בהיבט הציונים, נפעל בהיבט המכירות. אם מס' המכירות טרם המכירה הנוכחית היה שווה לאפס, אזי לא קיים צומת של הדגם בעץ המכירות ולכן פשוט נוסיף צומת זה. אחרת, נסיר את הצומת, נעדכן עבורו את מס' המכירות ונכניסו מחדש *לעץ המכירות*.

נדגיש כי לאחר כל שינוי *בעץ הציונים/המכירות/עץ האפסים הגדול/עץ האפסים הקטן* המתאים, נדאג לעדכן את המצביעים לצמתים המינימליים/מקסימליים של העצים הנ"ל.

<u>סיבוכיות זמן: $O(\log n + \log M)$- כאשר $M$ הוא מספר המודלים במערכת ו $n$ הוא מספר הסוגים.</u>

מציאת הצומת *בעץ הסוגים* נעשית בסיבוכיות $O(\log(n))$ (הבדיקה האם הסוג לא קיים נעשית באותה הפעולה). גישה למערכים כאשר הדגם נתון ועדכון הדגם הנמכר עבור הסוג נעשית במס' פעולות קבוע, כלומר בסיב' $O(1)$.

בהיבט ציונים – הסרה של צומת מעץ האפסים הקטן והכנסתו לעץ הציונים, נעשית בסיבוכיות של: $O(\log(n) + \log(m) + \log(M))$. כיוון שעלינו לאתר את הצומת המתאימה *בעץ האפסים הגדול* – ואז עלינו לבצע הסרות והכנסות בעצים המתאימים.

כיוון שמס' הדגמים הכולל גדול ממס' הדגמים של הסוג ובפרט מספרים אלו גדולים מ-1 (כמו גם מס' הסוגים), מתקיים:

$\log(m) \leq \log(M)$.

כלומר עבור $c = 2, n_0 = 1$ נקבל שלכל $M, n \geq n_0$ מתקיים:

$$\log(n) + \log(m) + \log(M) \leq \log(n) + 2 \cdot \log(M) \leq 2 \cdot (\log(n) + \log(M))$$

ולכן הסיבוכיות עבור חלק זה בפעולה היא $O(\log(n)+\log(M))$ - גם אם נדרשת הסרה של צומת הסוג *מעץ האפסים הגדול*. פעולה הפוכה בכיוונה, כלומר שבה מסירים צומת מעץ הציונים ומכניסים צומת לעץ האפסים הקטן, נעשית באותה סיבוכיות - שכן סיבוכיות הסרה זהה לסיבוכיות הכנסה בעץ AVL עם אותו מס' צמתים.
אם *עץ האפסים הקטן* לא מעורב, אז הסיבוכיות של הסרה והכנסה מחדש *בעץ הציונים* היא $O(\log(M))$ (שוב, עד כדי קבוע (2)).
גודלו המקסימלי של *עץ המכירות* כגודלו המקסימלי של *עץ הציונים* ומכאן שגם בו הכנסה / הסרה והכנסה נעשים באותה הסיבוכיות שבה הם נעשים בעץ הציונים. עדכון המצביעים לצמתים המינימליים/מקסימליים נעשה כאמור ב: $O(\log(n))$ או ב $O(\log(M))$.
סה"כ נקבל כי סיבוכיות הזמן של הפונקציה היא $\boldsymbol{O(\log(n)+\log(M))}$ כנדרש.

5 $\boldsymbol{StatusType}\ MakeComplaint(\boldsymbol{void}*DS,\ \boldsymbol{int}\ typeID, \boldsymbol{int}\ modelID, \boldsymbol{int}\ t)$

בעת הוספת תלונה למודל $modelID$ מהסוג $typeID$ נבצע את הפעולות הבאות:
נמצא את החולייה המתאימה ל$typeID$ בעץ הסוגים, משם ניגש למערך הציונים כדי להגיע לציון של המודל-$modelID$ ונעדכן את הציון שלו בהתאם לפרמטר-$t$ בעזרת הנוסחה : $grade-\left\lfloor\frac{100}{t}\right\rfloor$ .
בשלב הבא נרצה לעדכן את מקום החולייה שמתאימה ל$modelID$ בעץ הציונים המתאים לציון החדש, לשם כך נבצע את הבדיקות-

- <u>אם הציון של המודל לאחר התלונה הוא 0:</u>
  לפי מקרה זה נרצה להעביר את החולייה המתאימה ל-$modelID$ *מעץ הציונים לעץ האפסים הקטן*
  נמצא את $modelID$ *בעץ הציונים* בעזרת הציון הישן ונסיר אותו מהעץ.
  נחפש את $typeID$ ב-*עץ האפסים הגדול*.
  - אם $typeID$ לא נמצא בעץ משמע כל המודלים השייכים לסוג זה הם בעלי ציון השונה מאפס. במקרה זה נוסיף לעץ חוליה המתאימה ל$typeID$ .
  תחת $typeID$ *בעץ האפסים הגדול* שמור *עץ האפסים הקטן* של מודלים בעלי ציון אפס המתאימים לסוג זה - נוסיף *לעץ האפסים הקטן* את $modelID$.
- <u>אם הציון של המודל לפני התלונה הוא 0:</u>
  לפי מקרה זה נרצה להעביר את החולייה המתאימה ל$modelID$ מעץ אפסים הקטן לעץ הציונים ההחלפה מתבצעת באופן דומה למקרה הקודם.
  - אם לאחר הסרת המודל עץ האפסים הקטן התרוקן - נמחק את $typeID$ מעץ האפסים הגדול.
- <u>אחרת:</u>
  הציון לפני ואחרי התלונה שונים מ-0, ולכן שינוי מקום החולייה המתאימה ל-$modelID$ מתבצע בעץ הציונים (ולא מערב את עץ האפסים).

בשני המקרים הראשונים- נעדכן את המצביע לחוליה המינימלית בעץ האפסים הגדול ונעדכן את המשתנה המקומי של מספר המודלים בעלי ציון 0 במערכת בהתאם למקרה המתאים:
$zero\_graded\_models\_num++\ /\ zero\_graded\_models\_num--$
בשלושת המקרים נעדכן את המצביע לחולייה המינימלית בעץ המודלים בעלי הציון.

<u>סיבוכיות זמן: $O(\log n+\log M)$- כאשר $M$ הוא מספר המודלים במערכת ו $n$ הוא מספר הסוגים.</u>

1. סיבוכיות הזמן בעבור מציאת חוליה בעצי הסוגים ובעץ הציונים : בהתאמה $O(\log n)$ ו- $O(logM)$.
   (הבדיקה האם הסוג קיים נעשית בפעולת החיפוש).
   בעבור עץ האפסים הקטן : $O(\log m)$ כאשר $m$ הוא לכל היותר מספר המודלים ששייכים לסוג המבוקש, מאחר ומתקיים $m<M$ ולכן $\log m<logM$ נוכל להזניח גודל זה. (לפי אותו ההסבר שתואר בפונקציית $SellCar$).
2. הסרה והוספה של חוליה לעצי הסוגים ולעץ הציונים בעזרת אלגוריתמי הגלגול שנלמדו בהרצאה ומומשו בקוד בסיבוכיות זמן- $O(\log n)$ ו- $O(logM)$ בהתאמה. משיקולים דומים נזניח את סיבוכיות ההכנסה וההוצאה מעץ האפסים הקטן.
3. עדכון הציון (גישה למערך המקום ידוע) ועדכון המצביעים הן פעולות של $O(1)$

סה"כ נקבל שסיבוכיות הזמן מסתכמת ב: $\boldsymbol{O(\log M+\log n)}$ כנדרש.

6 $\boldsymbol{StatusType}\ GetBestSellerModelByType(\boldsymbol{void}*DS,\ \boldsymbol{int}\ typeID, \boldsymbol{int}*modelID)$

- <u>אם $typeID=0$:</u>
  אנחנו נדרשים להחזיר את הדגם בעל מספר המכירות הגדול ביותר בכל המערכת.

לשם כך נעזר במצביע לחולייה המקסימלית בעץ המכירות- נחזיר את מספר המודל של החולייה הזו.
- אם עץ המכירות ריק נחזיר את מספר המודל 0, שכן אף רכב לא נמכר ולכן לפי דרישות התרגיל יש להחזיר את הרכב בעל מספר המודל הקטן ביותר.

- אחרת-

ניגש ל-$typeID$ בעץ הסוגים ונחזיר את ערך המשתנה של המודל הנמכר ביותר בסוג, השמור תחת החולייה של $typeID$.

סיבוכיות זמן: $O(\log n)$:

גישה למצביע ששמור במבנה הנתונים זו פעולה שנעשית ב $O(1)$.
בדיקה אם העץ ריק נעשית גם היא ב $O(1)$. (בדיקה האם המצביע לשורש הוא $nullptr$)
גישה לחוליה עם מפתח ידוע בעץ הסוגים שגודלו n היא $O(\log n)$.
סה"כ נקבל כי סיבוכיות הזיכרון של הפונקציה היא $\boldsymbol{O(\log n)}$ כנדרש.

**7** $\boldsymbol{StatusType}\ GetWorstModels(\boldsymbol{void} * DS,\ \boldsymbol{int}\ numOfModels\ ,\boldsymbol{int} * types, \boldsymbol{int} * models)$

את השמות המודלים (למערכים) לפי סדר הציונים נעשה באמצעות סיורי in order עם שינויים קלים, בעץ הציונים ובעצי האפסים הקטנים.
על מנת לעמוד בדרישות הסיבוכיות של התרגיל נתחיל את הסיורים מהחולייה המינימלית של כל עץ (שנגישה לנו ב$O(1)$ בעזרת מצביעים שהגדרנו במבנה הנתונים)

נגדיר משתנה בוליאני- predecessor_negative שיאותחל בfalse. הוא יתפקד כדגל שנותן אינדיקציה לגבי סימן החולייה בה ביקרנו לפני החולייה הנוכחית.
נגדיר counter שיאותחל ב-0, בכל השמה שנבצע נקדם את הcounter, כך נדע לעצור אחרי שביצענו $numOfModels$ השמות.

- אם עץ הציונים ריק או שהחולייה המינימלית שייכת למודל עם ציון חיובי ניגש לעץ האפסים הגדול על מנת "לקחת" את המודלים הנמצאים שם (בצמתים בעצי האפסים הקטנים).
נתחיל בסיור in order בעץ האפסים הגדול מהחולייה המינימלית. בכל חולייה שנגיע אליה בסיור ניכנס לעץ האפסים הקטן ששמור בה ונבצע סיור in order פנימי גם כן מהחולייה המינימלית. (אם עץ הציונים לא ריק ולא בצענו $numOfModels$ השמות נמשיך בסיור inorder בעץ הציונים כפי שיתואר בהמשך).
- אחרת, קיימים בעץ הציונים מודלים בעלי ציון שלילי, מכאן שנרצה להכניס אותם ראשונים למערכים.
נשנה את הדגל שהגדרנו ל-true ונכניס למערכים תוך כדי סיור inorder את המודלים בעליי הציון השלילי, נמשיך כך כל עוד לא הגענו ל$numOfModels$ והחולייה הנוכחית בסיור שייכת למודל עם ציון שלילי.
- אם הגענו לחולייה בעלת ציון חיובי נשנה את הדגל לfalse שוב ונבצע את הבדיקות:
  - אם קיימים מודלים בעליי ציון 0 (אינדיקציה שנוכל לקבל ב$O(1)$ בעזרת המשתנה המקומי שסופר את מספר המודלים בעלי ציון 0 במערכת). נעבור לסיורי in order בעץ האפסים הגדול ובעצי האפסים הקטנים כפי שמתואר לעיל. אם לא בצענו $numOfModels$ הדפסות נמשיך לחלק הבא.
  - נמשיך בסיור inorder בעץ הציונים מהנקודה בה עצרנו ונמשיך עד שנבצע $numOfModels$ הדפסות.

סיבוכיות זמן: $O(m)$- כאשר m הוא מספר הדגמים המבוקש- $numOfModels$

מאחר ובכל סיורי ה-in order שבצענו התחלנו מהחולייה המינימלית, הקריאות הרקורסיביות נעשות רק על הבן הימני של החולייה הנוכחית בסיור. בדרך זו אנחנו סורקים את אבריי העץ מהקטן לגדול כאשר מתחילים מהמינימום ואז עוברים לסיור in order על תת העץ הימני שלו (בלבד, ניתן לעשות זאת כי התחלנו מהמינימלי ולכן בקריאה הרקורסיבית הראשונה- אין תת עץ שמאלי, ובקריאות הבאות ניכנס לרקורסיה עם האבא של הצומת הקודמת ומאחר והתחלנו מהמינימלי- תתי העצים שכבר עברנו עליהם ימצאו בתת העץ השמאלי של האבא ומכאן שאנחנו לא מפספסים אף צומת).
מאחר ולא התחלנו בשורש (כמו במימוש הרגיל של in order) חסכנו את הפעולה של לרדת בעץ עד לאיבר המינימלי תוך כדי הסיור, פעולה שעולה $O(\log n)$ כאשר $n$ הוא גודל העץ. ולכן סיור ה- in order לפי המימוש שלנו הוא $O(m)$ (לפי הנחיות התרגיל אין צורך להוכיח שסיור in order לאיבר ה-$i$ לוקח $O(i)$ זמן).
שאר הפעולות של הפונקציה- עדכון המשתנה הבוליאני וה-$counter$ לוקחים $O(1)$.
בנוסף גישה לחוליות המינימליות בכל עץ גם כל לוקחת $O(1)$ שכן מצבעים אלו שמורים במבנה הנתונים שלנו.
סה"כ נקבל שסיבוכיות הזמן מסתכמת ב: $\boldsymbol{O(m)}$ כנדרש.

**8** $\boldsymbol{void}\ Quit(\boldsymbol{void} ** DS)$

בעת הקריאה ל-$Quit$ נבקש לשחרר את ההקצאות שביצענו ב-$Init$ ובביצוע הפעולות שתוארו לעיל.
ראשית נעדכן את 3 המצביעים במבנה ל$nullptr$ .

נהרוס את עץ הציונים, ואת עץ המכירות.
נסייר ב inorder על חוליות עץ האפסים הגדול, בכל חולייה כזו נעדכן את המצביעים ששמורים בה ל $nullptr$
ונהרוס את תת-עץ האפסים הקטן ששמור בה. בסוף הסיור נהרוס את עץ האפסים הגדול.
נסייר ב inorder על חוליות עץ האפסים הראשי ובכל חולייה נהרוס את 2 המערכים ששמורים בה.
בסוף הסיור נהרוס את עץ הסוגים הראשי.
<u>סיבוכיות זמן:$O(m+n)$- כאשר $m$ הוא מספר הדגמים במערכת ו $n$ הוא מספר הסוגים במערכת.</u>
הריסת כל עץ נעשית בסיור post order שמתבצע בסיבוכיות $O(n)$ בעבור עצי הסוגים ו$O(m)$ בעבור עצי המודלים.
עדכון המצבעים ל $nullptr$ מתבצע ב$O(1)$.
סה"כ נקבל שסיבוכיות הזמן מסתכמת ב: $\boldsymbol{O(m+n)}$ כנדרש.

**עמידה בסיבוכיות המקום עבור המבנה והפעולות לעיל:**

במהלך ייצור ותחזוקת העצים במערכת מתקיים:

מספר האיברים בעץ הסוגים ובעץ האפסים הגדול תהיה לכל היותר $n$ כאשר $n$ הוא מספר הסוגים במערכת סה"כ $2n$ בעבור שני העצים.

בעץ המכירות, הציונים וסך כל עצי האפסים הקטנים יהיו לכל היותר $m$ אברים כאשר $m$ הוא מספר המודלים במערכת (נבחין כי לא ייתכנו $m$ אברים גם בעץ הציונים וגם בסך כל עצי האפסים קטנים- אך נחשב את סיבוכיות המקום לחומרה תוך התעלמות מפרט זה). סה"כ $3m$ לעצים אלו.

בכל חוליה בעץ הסוגים יש שני מערכים שגודלם כמספר המודלים השייכים לסוג זה ולכן סיבוכיות המקום עבור סך כל המערכים בעץ הסוגים תהיה $m$ לכל מערך ו$2m$ לשני המערכים.

בכל חוליה בעץ האפסים הגדול יש מצביע לצומת המינימלי בעץ האפסים הקטן שבצומת זה, לכל מצביע סיבוכיות מקום של $O(1)$ ועבור כל המצבעים הוקצה $n$ מקום כמספר החוליות בעץ האפסים הגדול.

בנוסף מספר קבוע של מצביעים ומשתנים מקומיים שהוקצו במבנה להם סיבוכיות מקום של $O(1)$.

סה"כ קבלנו שהקצאת המקום עבור המערכת, לאורך כל שלבי התחזוקה שלה היא לכל היותר

$$const \cdot (n+m)$$

ומכן שסיבוכיות המקום היא $\boldsymbol{O(n+m)}$ כנדרש.