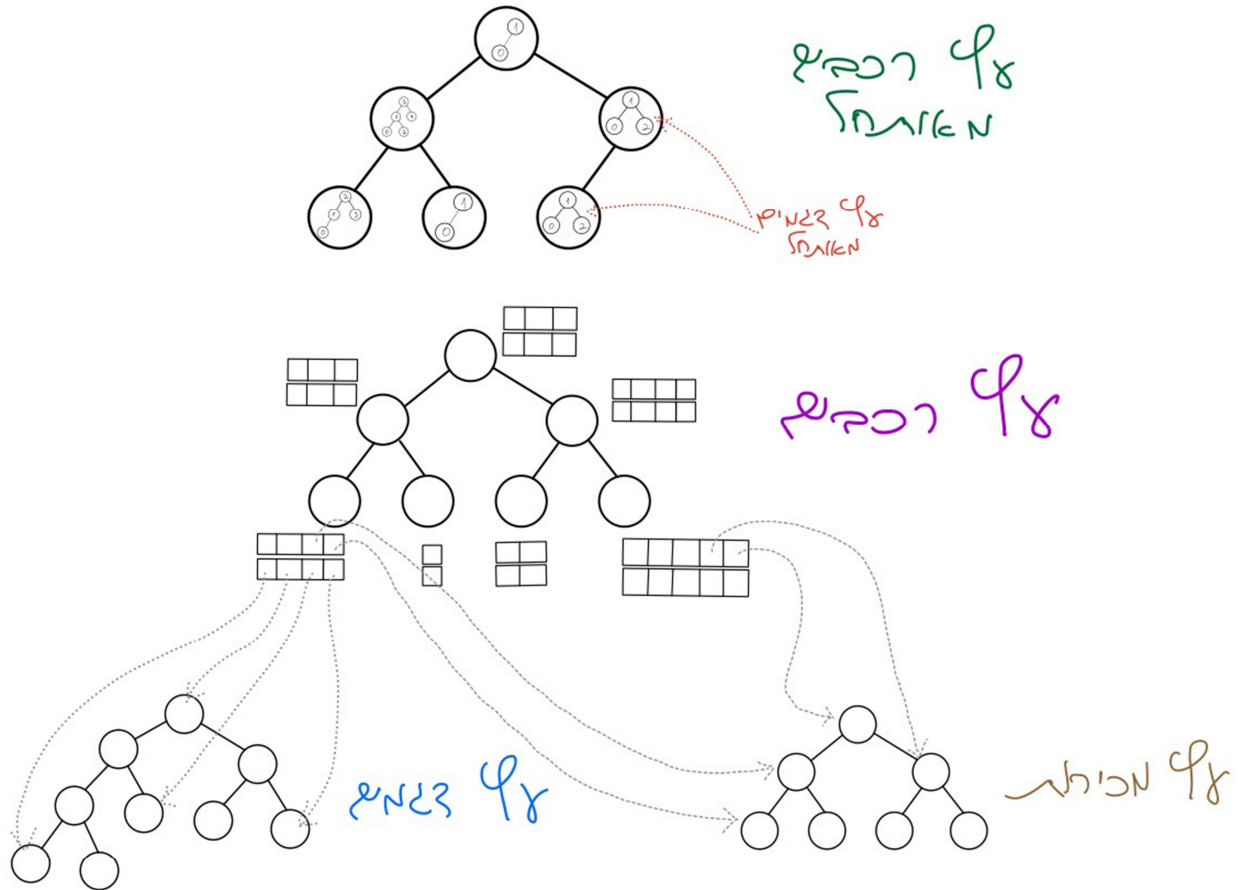


גיליון רטוב 1

סרטוט המבנה:



הסבר המבנה:

מבנה הנתונים יכול 4 עצים מרכזיים:

1. עץ AVL של סוגי הרכבים - עץ רכבים.
 2. עץ AVL של סוגי רכבים שיש להם דגמי רכבים שטרם נמכרו - עץ רכבים מאותחל.
כך שכל סוג רכב מכיל עץ סוגי דגמים שלו שטרם נמכרו - עץ דגמים מאותחל.
 3. עץ AVL של דגמי הרכבים שנמכרו - עץ דגמים.
 4. עץ AVL של מספר מכירות לפי סוג רכב - עץ מכירות.
- *הצבעים בהתאם לסרטוט.

כל איבר בעץ הרכבים המאותחל יכול:

1. מספר המציין סוג רכב - העץ ממזין לפי משתנה זה.

2. עץ כמעט שלם של סוגי הדגמים של הרכב הנוכחי שטרם נמכרו (עץ דגמים מאותחל) - עץ זה ממזין לפי סוג הדגם.

כל איבר בעץ הרכבים יכיל:

1. מספר המציין סוג רכב - העץ ממזין לפי משתנה זה.
2. מצביע לאיבר בעץ המכירות המייצג את סוג רכב.
3. מערך מצביעים בגודל מספר הדגמים של הרכב, המצביעים אל האיבר המתאים בעץ דגמים.
4. מערך מצביעים בגודל מספר הדגמים של הרכב, המצביעים אל האיבר המתאים בעץ המכירות.
5. מספר המודל עם מספר המכירות המקסמלי.

כל איבר בעץ המכירות יכיל:

1. מספר המכירות הגדול ביותר מבין הדגמים השונים עבור סוג הרכב הנוכחי - העץ ממזין לפי משתנה זה בעדיפות 1.
2. מספר המציין סוג רכב - העץ ממזין לפי משתנה זה בעדיפות 2.
3. מספר המציין את סוג הדגם עבורו מספר המכירות הוא הגדול ביותר ביחס לסוג הרכב הנוכחי - העץ ממזין לפי משתנה זה בעדיפות 3.

כל איבר בעצי הדגמים (המאותחל והרגיל) יכיל:

1. מספר מכירות.
2. ציון הדגם - העץ ממזין לפי משתנה זה בעדיפות 1.
3. מזהה סוג הרכב - העץ ממזין לפי משתנה זה בעדיפות 2.
4. מזהה הדגם - העץ ממזין לפי משתנה זה בעדיפות 3.

כל עץ יוגדר ע"י מחלקה.

הערות:

הוצאת איבר מתוך עץ AVL בעל n איברים מבלי להרוס את האיזון - סיבוכיות זמן $O(\log n)$
(נלמד בהרצאות)

האלגוריתם הכללי וחישוב סיבוכיות זמן:

void* Init()

פעולות	סיבוכיות זמן
נבנה 4 עצים ריקים.	$O(1)$

סיבוכיות זמן סה"כ: $O(1)$

StatusType AddCarType (void *DS,int typeID, int numOfModels)

פעולות	סיבוכיות זמן
נבנה איבר חדש עבור עץ המכירות - כולל אתחול מערך באורך m	$O(m)$
נבנה איבר חדש עבור עץ הרכבים - כולל אתחול מערך באורך m	$O(m)$
נכניס את האיבר המתאים לעץ הרכבים.	$O(\log n)$
נכניס את האיבר המתאים לעץ המכירות.	$O(\log n)$
*נבנה את תת עץ הדגמים המאותחל.	$O(m)$
נבנה איבר חדש עבור עץ הרכבים המאותחל.	$O(1)$
נכניס את איבר סוג רכב זה לעץ הרכבים המאותחל.	$O(\log n)$

*עץ הדגמים המאותחל יבנה כך:
לקיחת האיבר האמצעי והגדרתו כשורש.
הפעלת אותה פעולה על חצי מערך ימני וחצי מערך שמאלי.

סיבוכיות זמן סה"כ: $O(\log n + m)$

StatusType RemoveCarType(void * DS, int typeID)

פעולות	סיבוכיות זמן
נחפש את האיבר המתאים בעץ הרכבים.	$O(\log n)$
בעזרת מערך המצביעים נמצא את האיברים להסרה מעץ הדגמים. עבור כל איבר - נקרא להסרה שלו מהעץ.	$O(m \log M)$
בעזרת מערך המצביעים נמצא את האיברים להסרה מעץ המכירות. עבור כל איבר - נקרא להסרה שלו מהעץ.	$O(m \log M)$
נסיר את האיבר המתאים מעץ הרכבים.	$O(\log n)$
נחפש את האיבר המתאים בעץ הרכבים המאותחל.	$O(\log n)$
נמחק את עץ הדגמים המאותחל המוכל בו.	$O(m)$
נסיר את איבר זה מעץ הרכבים המאותחל.	$O(\log n)$

סיבוכיות זמן סה"כ: $O(\log n + m \log M)$

הוכחה:

$$\log n + m \log M + m \leq \log n + m \log M + m \log M =$$

$$= \log n + 2 m \log n = O(\log n + m \log m)$$

StatusType SellCar(void * DS, int typeId, int modelID)

סיבוכיות זמן	פעולות
$O(\log n)$	נחפש את סוג הרכב המתאים בעץ סוגי הרכבים המאותחל.
$O(\log M)$	נחפש את סוג הדגם המתאים בעץ סוגי הדגמים המאותחל שבתוך סוג רכב זה.
$O(1)$	נסיר את הדגם המתאים מעץ הדגמים המאותחל (אם קיים שם).
$O(\log n)$	נחפש את סוג הרכב בעץ הרכבים.
$O(\log n)$	נכניס/נעדכן את הדגם המתאים בתוך עץ הדגמים בעץ הרכבים באמצעות הוצאה של האיבר הישן והכנסה של החדש.
$O(\log n)$	נכניס/נעדכן את הדגם המתאים בעץ המכירות באמצעות הוצאה של האיבר הישן והכנסה של החדש.

סיבוכיות זמן סה"כ: $O(\log n + \log M)$

*בעץ סוגי הרכבים המאותחל - אם כל אחד מסוגי הדגמים של סוג רכב מסוים נמכרו, אז האיבר המייצג את סוג הרכב בעץ זה יוסר.

StatusType MakeComplaint(void * DS, int typeId, int modelID, int t)

סיבוכיות זמן	פעולות
$O(\log n)$	נחפש את האיבר המתאים בעץ הרכבים.
$O(1)$	ניגש אל האיבר המתאים בעץ הדגמים (יש לנו מצביע אליו).
$O(1)$	נעדכן את ציון הרכב - לפי מספר החודשים שנתון כפרמטר.
$O(\log M)$	נוציא את האיבר מהעץ
$O(\log M)$	נכניס את האיבר בחזרה לעץ

סיבוכיות זמן סה"כ: $O(\log M + \log n)$

StatusType GetBestSellerModelByType(void *DS, int typeId, int *modelID)

סיבוכיות זמן	פעולות
$O(\log n)$	נחפש את האיבר המתאים בעץ הרכבים.
$O(1)$	ניגש אל האיבר המתאים בעץ המכירות (יש לנו מצביע אליו דרך המערך ומספר הדגם המקסימלי שמתעדכן בכול - sellCar).
$O(1)$	נחזיר מצביע לסוג הדגם.

סיבוכיות זמן סה"כ: $O(\log n)$

*בנוסף מבנה עץ הרכבים יחזיק מצביעים של האיבר הכי גדול והכי קטן שמתעדנים בכול הוצאה והכנסה כך במקרה שבמקרה בו $typeID=0$ אז הסיבוכיות היא $O(1)$.

StatusType GetWorstModels(void *DS,int numOfModels, int * types, int * models)

סיבוכיות זמן	פעולות
$O(m)$	נסרוק עד m איברים מעץ הדגמים inorder לתוך מערך בגודל m
$O(m)$	נסרוק עד m דגמים מעץ הרכבים המאותחל inorder לתוך מערך בגודל m
$O(m)$	נאחד את המערכים למערך אחד באורך $2m$
$O(m)$	קריאה של m איברים ראשונים במערך המיוחד והחזרת הערכים

סיבוכיות זמן סה"כ: $O(m)$

*סריקה של m איברים מתוך עת הרכבים המאותחל נעשית באמצעות סריקה inorder של העץ עצמו וסריקת כל איבר בעץ inorder.

*בטבלה מתואר המימוש האידאלי - נציין כי במימוש שלנו סיבוכיות זמן ש inorder היא $O(m + \log n)$ ולכן סה"כ סיבוכיות הזמן היא $O(m + \log n)$

Void Quit(void **DS)

סיבוכיות זמן	פעולות
$O(m)$	מחיקת עץ הדגמים.
$O(n)$	מחיקת עץ המכירות.
$O(m)$	מחיקת מערך דגמים הנמצא באיבר בעץ המכירות.
$O(n)$	מחיקת עץ הרכבים.
$O(m)$	מחיקת עץ דגמים מאותחל הנמצא בעץ הרכבים המאותחל.
$O(n)$	מחיקת עץ הרכבים המאותחל.

סיבוכיות זמן סה"כ: $O(n + m)$

*במקרה של quit המחיקה נעשית בצורה עיוורת ללא חיפוש בעץ ולכן בסיבוכיות $O(x)$ כאשר x הוא מספר האיברים שנמחקים.

חישוב סיבוכיות מקום:

סיבוכיות המקום שלו	מבנה הנתונים
$O(n) \rightarrow n$ איברים	עץ סוגי הרכבים
$O(m) \rightarrow m$ איברים	סכום אורכי המערכים בתוך עץ סוגי הרכבים
$O(m) \rightarrow m$ איברים	עץ הדגמים הכולל

סה"כ: $O(n + m)$

כל הקצאות הזיכרון נעשות בגודל כפולה של m או n או מספר חסום מלעיל על ידי m או n ולכן נקבל בסה"כ מספר חסום על ידי קומבינציה לינארית של $m+n$.

נוכיח גם עבור מקרים "בעייתיים":

בעץ הדגמים המאותחל יש x איברים וכל אחד מהם מכיל תת עץ עם y_i איברים.

מתקיים $x \leq n$ ו $y_1 + \dots + y_n = m$ ולכן:

$$\sum_1^x y_i \leq \sum_1^n y_i = y_1 + \dots + y_n = m$$