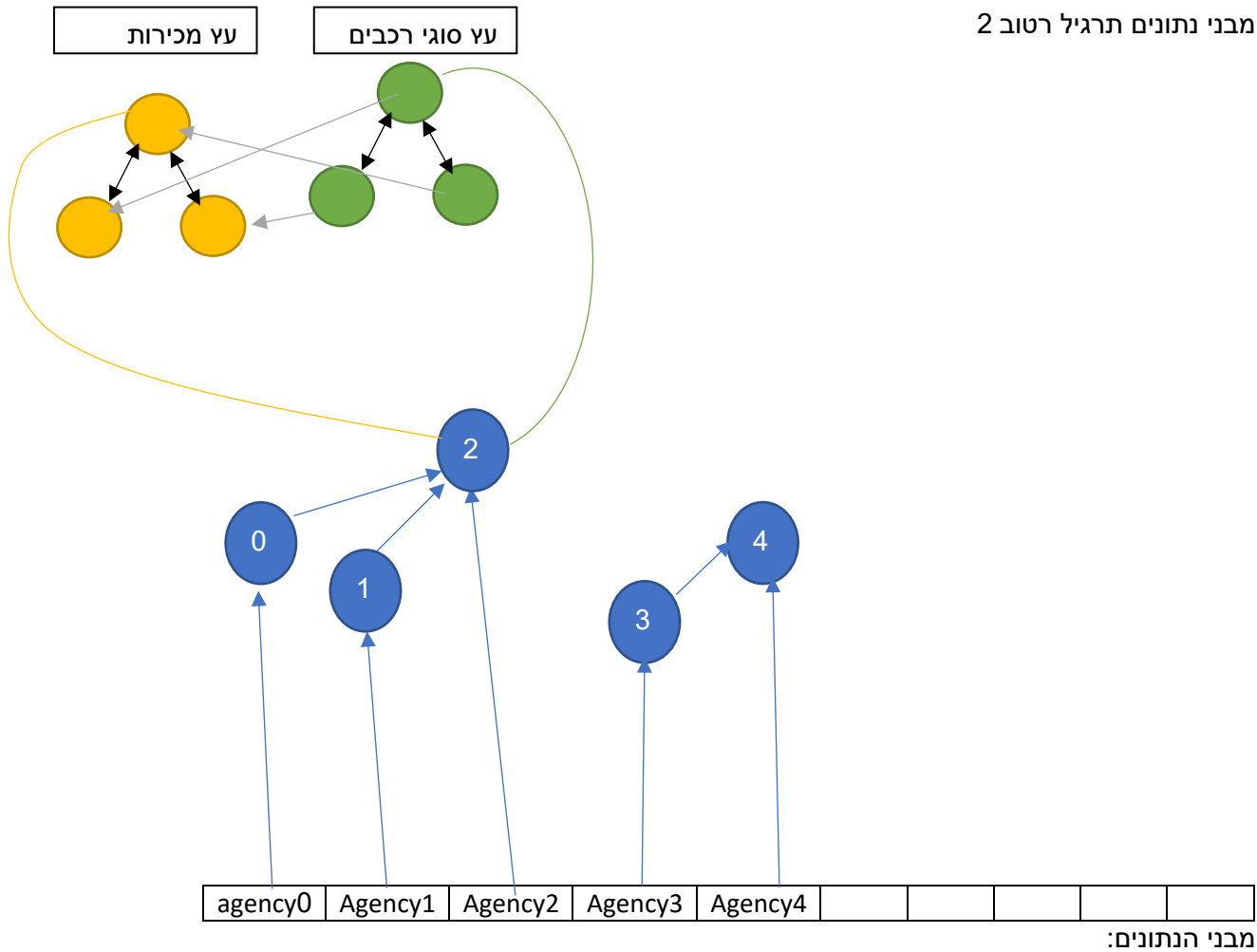


מבני נתונים תרגיל רטוב 2



נשתמש ב-UF עם שינויים קלים כאשר ה-UF מורכב באופן הבא:

מערך דינמי של מצביעים ל עץ הפוך שכל node שבו מסמל סוכנות שנוספה למערכת וכל node-ים בעץ הם סוכנויות שהתאחדו ומייצגים סוכנות אחת מאוחדת כל node יכול:

דרגה מספר הצמתים בתת העץ היוצא מצמת זה כשורש כולל צמת זה.

מזהה הסוכנות -כפי שהוזנה במערכת תחילה לפני שאוחדה עם סוכנויות אחרות בעץ.

מצביע להורה שיסייע בפעולות השונות בעץ.

בפרט השורש של העץ ההפוך יכיל שני עצי דרגות היוצאים ממנו:

1. עץ מכירות-עץ AVL שהוא גם עץ דרגות שכל node מכיל מזהה של סוג הרכב שנמכר בעבר ומספר המכירות שלו -עץ זה ממזין לפי מספר המכירות הגבוהה ראשית ומשנית לפי מזהה סוג הרכב הגבוהה.

2. עץ סוגי רכב- עץ AVL שכל node מכיל מזהה של סוג הרכב שנמכר בעבר ומצביע לצמת בעץ המכירות המכיל את אותו המזהה סוג רכב ממזין לפי מזהה סוג הרכב

נחזיק משתנה הסופר את גודל המערך הדינמי ומשתנה הסופר את מספר התאים במערך שבשימוש.

כנראה בהרצאה נשתמש האיחוד קבוצות לפי גודל ובדחיסה של מסלול בקריאה לfind כדי לשמור על סיבוכיות הזמן הדרושה שתפורט בהמשך.

```
void * Init()
```

נאתחל את מבנה הנתונים.

נאתחל מערך ריק בגודל 1.

נעדכן את המשתנה של גודל המערך ל1

ואת מספר התאים בשימוש ל0

```
StatusType AddAgency(void *DS)
```

נבדוק תקינות קלט ונחזיק שגיאה במידת הצורך.

נבדוק האם גודל המערך הנוכחי שווה למספר התאים בשימוש (כלומר המערך מלא).

אם כן נקצה מערך חדש עם גודל כפול 2 ונעביר לשם את כל תכן התאים מהמערך הישן ונמחק את המערך הישן.

נוסיף תא חדש המצביע לnode חדש במערך במקום ששווה למספר התאים בשימוש במערך (שזה שווה למספר הסוכניות שנוספו למערכת) נגדיל את המשתנה שסופר את מספר התאים בשימוש באחד.

ה node החדש מייצג את הסוכנות שנוספה ומכיל את המזהה של הסוכנות (מזהה הסוכנות שווה למשתנה מספר התאים בשימוש לפי שהגדלנו אותו באחד).

סיבוכיות זמן משוערכת:

סיבוכיות הזמן המשוערכת של תחזוק מערך בגודל דינמי היא $O(1)$ בפרט מערך דינמי המשתמש רק בהגדלות בנוסף עשינו מספר פעולות בזמן קבוע במקרה הגרוע שבפרט זהו גם הזמן המשוערך שלהם ולכן בסך הכל סיבוכיות הזמן המשוערכת היא $O(1)$

```
StatusType SellCar(void *DS, int agencyID, int typeID, int k)
```

נבדוק תקינות קלט ואם רכב זה קיים במערכת ונזרוק שגיאה במידת הצורך.

כעת נבצע חיפוש של הסוכנות אם זהו מזהה של סוכנות שאוחדה בעבר עם אחרות נעלה לשורש העץ ההפוך המייצג את הסוכנות המאוחדת אם לא אוחדה השורש זה ה node הראשון שיוצא מהמצביע שבמערך.

כעת מתוך עץ סוגי הרכבים היוצא מן השורש של העץ ההפוך נחפש את סוג רכב ונפצל למקרים הבאים:

אם נמצא סוג הרכב כלומר רכב זה נמכר בעבר ניגש למצביע הנמצא בצמת בעץ סוגי הרכבים שמצביעה על הצמת התואם בעץ המכירות נמחק את הצמת עם מספר המכירות הלא מעודכן מעץ המכירות ונוסיף בעץ המכירות צמת חדש עם מספר מכירות מעודכן (תוספת של k מכירות למספר המכירות הקודם), נעדכן את המצביע מהצמת בעץ הסוגים להצביע על הצמת החדש בעץ המכירות.

אם לא נמצא סוג הרכב כלומר זאת הפעם הראשונה שבא הוא נמכר נוסף צמת עם K מכירות ומזהה סוג הרכב לעץ המכירות וצמת לעץ הסוגים עם מזהה סוג הרכב שמצביע לצמת שהוספנו בעץ המכירות.

סיבוכיות זמן משוערכת:

הפעולות שבוצעו ב uf הם חיפוש בלבד, בעוד שבפונקציות `GetlthSoldType` `UniteAgencies` השתמשנו באיחוד ובחיפוש במספר קבוע של פעמיים לפי ההרצאה הסיבוכיות המשוערכת היא $O(\log^* n)$ כאשר n הוא מספר הסוכנויות במערכת כלומר הצמתים ב uf .

באשר לפעולות שבוצעו בעצי החיפוש במקרה בגרוע ביצענו מחיקה והוספה לשני העצים בעלות של $O(\log m)$ במקרה הגרוע ולכן בפרט $O(\log m)$ במשוערך כאשר m הוא מספר סוגי הרכבים בסוכנות במערכת לכן יש $2m$ צמתים בעצים של הסוכנות

על כן ס"כ הסיבוכיות המשוערכת היא $O(\log m + \log^* n)$

`StatusType UniteAgencies(void *DS, int agencyID1, int agencyID2)`

נבדוק את תקינות הקלט

נפעיל איחוד על שתי הסוכנויות המאוחדות ש $agency1/2$ שייכות אליהם כנראה בכיתה כאשר מערך המצביעים ישאר כפי שהיה כלומר עדיין התאים התואמים את הסוכנויות מצביעים ל `node` הראשוני שנוצר כאשר הסוכנות $agency1/2$ נוספה לראשונה שכעת הוא חלק מסוכנות המאוחדות שכוללת את כל הסוכנויות שאוחדו עם $agency1/2$.

נאחד גם את העצי המכירות ואת עצי סוגי הרכבים ב $O(m_1 + m_2)$ כאשר m_1 הוא מספר הרכבים ב $agency1$ ו m_2 הוא מספר הרכבים ב $agency2$ לפי האלגוריתם הנראה בתרגול (***הערה על כך** **בהמשך**). כאשר נמחק את העצים הישנים והעצים החדש יהיו שייכים לקבוצה הגדולה מבין הקבוצה המאוחדת של $agencyID1$ או לקבוצה המאוחדת של $agencyID2$. במילים אחרות עץ המכירות המאוחד ועץ סוגי הרכבים המאוחד החדשים יצאו מתוך השורש של העץ ההפוך החדש שמייצג את הסוכנות המאוחדת.

סיבוכיות זמן משוערכת:

הפעולות שבוצעו ב uf הם איחוד בלבד, בעוד שבפונקציות `GetlthSoldType` `SellCar` השתמשנו בחיפוש במספר קבוע של פעמיים לפי ההרצאה הסיבוכיות המשוערכת היא $O(\log^* n)$ כאשר n הוא מספר הסוכנויות במערכת כלומר הצמתים ב uf .

באשר לפעולות שבוצעו בעצי החיפוש ביצענו איחוד לשני עצים פעמיים בעלות של $2O(m_1 + m_2)$ מכאן כי בפרט זאת סיבוכיות הזמן המשוערכת.

על כן ס"כ הסיבוכיות המשוערכת היא $O(m_1 + m_2 + \log^* n)$

***ליתר דיוק** – אמנם בתרגול האלגוריתם דיבר על שני עצים בגודל זהה n

אך במקרה של עצים בגדלים שונים (m_1, m_2) זה באותו אופן יעבוד בזמן של $O(m_1 + m_2)$ פירוט במידת הצורך:

סיוור *Inorder* על העצים והדפסתם לתוך מערך $O(m_1 + m_2)$

מיזוג המערכים באמצעות פונקציית $merge$ $O(m_1 + m_2)$

אתחול עץ ממערך לפי אלגוריתם אתחול עץ כמעט שלם ב $O(m_1 + m_2)$ כגדול המערך הממוזג במקרה של עץ דרגות נעשה סיור $postorder$ לעדכון הדרגות כנל לגבהים בכל צמת. $O(m_1 + m_2)$

סהכ זמן

$O(m_1 + m_2)$.

StatusType GetIthSoldType(void *DS, int agencyID, int i, int* res)

נבדוק תקינות קלט ושקיימת סוכנות זאת במבנה שלנו עם דרגת שורש שלא קטנה מ/ ונזרוק שגיאה בהתאם.

נחפש את השורש של העץ ההפוך המייצג הסוכנות המאוחדת של agencyID ניגש לעץ המכירות היוצא ממנו (זהו עץ דרגות) ונפעיל את אלגוריתם $select(i)$ כפי הנראה בתרגול.

נחזיר את סוג הרכב שנמצא באלגוריתם ב res

סיבוכיות זמן משוערכת:

הפעולות שבוצעו ב uf הם חיפוש בלבד, בעוד שבפונקציות UniteAgencies SellCar השתמשנו בחיפוש במספר קבוע של פעמיים לפי ההרצאה הסיבוכיות המשוערכת היא $O(\log^* n)$ כאשר n הוא מספר הסוכנויות במערכת כלומר הצמתים ב uf .

באשר לפעולות שבוצעו בעצי החיפוש ביצענו את אלגוריתם $select(i)$ כפי הנראה בתרגול בסיבוכיות זמן מקרה גרוע של $\log m$

מכאן כי בפרט זאת סיבוכיות הזמן המשוערכת.

על כן ס"כ הסיבוכיות המשוערכת היא $O(\log m + \log^* n)$

void Quit(void **DS)

נעבור על המערך של המצביעים בכל תא שאותחל ונמחק את $node$ אליו הוא מצביע אם $node$ זה הוא גם שורש (נדע זאת כי העצים היוצאים ממנו מאותחלים) נמחק את העצים היוצאים ממנו לפי סדר $Postorder$ כדי לשמר את הסיבוכיות של :

$O(\text{number vertex in tree})$

ולאחר מכן נמחק $node$ זה.

נמחק את המערך ואת מבנה הנתונים ונעדכן את DS ל $NULL$

בהתאם לגודל המבנה בכל שלב במקרה הגרוע המפורט למטה בסיבוכיות המקום :

מחיקת כלל העצים = $O(\text{number vertex in tree})$ לכל העצים במערכת סהכ $2O(m)$

מעבר על המערך ומחיקת כל $Node$ אליו הוא מצביע מחיקת המערך $O(n)$

מחיקת המערך שגודלו יכול להיות כפול 2 ממספר הסוכניות לכל היותר $2O(n)$

סהכ:

$$3O(n) + 2O(m) = O(n + m)$$

סיבוכיות מקום:

בכל שלב אנחנו מחזקים במקרה הגרוע מערך בגודל מספר הסוכניות שהוספו למערכת כפול 2

$$2O(n)$$

$Node$ של העצים ההפוכים כמספר הסוכניות שהוספו למערכת. $O(n)$

2 עצים בגודל כולל של מספר סוגי הרכב במערכת $2O(m)$

סהכ:

$$3O(n) + 2O(m) = O(n + m)$$