

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 1 מתוך 8



עידו גליל, [idoغالil@cs.technion.ac.il](mailto:idoغالil@cs.technion.ac.il)

**מתרגל ממונה על התרגיל:**

23:55 בשעה 13/06/2021

**תאריך ושעת הגשה:**

בזוגות. יורד ציון לתרגילים שיוגשו ביחידים בלי אישור מהמתרגל הממונה על

**אופן ההגשה:**

התרגיל.

**הנחיות כלליות:**

- תשובות לשאלות המרכזיות אשר ישאלו יתפרסמו ב-FAQ לטובת כלל הסטודנטים. שימו לב כי **תוכן ה FAQ הוא מחייב וחובה לקרוא אותו**, אם וכאשר הוא יתפרסם. **לא** יתקבלו דחיות או ערעורים עקב אי קריאת ה FAQ.
- לפני שאתם ניגשים לקודד את פתרוןכם, ודאו כי יש לכם פתרון העומד **בכל** דרישות הסיבוכיות התרגיל. תרגיל שאינו עומד בדרישות הסיבוכיות יחשב כפסול.
- **העתקת תרגילי בית רטובים תיבדק באמצעות תוכנת בדיקות אוטומטית, המזהה דמיון בין כל העבודות הקיימות במערכת, גם כאלו משנים קודמות. לא ניתן לערער על החלטת התוכנה. התוכנה אינה מבדילה בין מקור להעתק! אנא הימנעו מהסתכלות בקוד שאינו שלכם.**

# מבני נתונים 234218 אביב תשפ"א

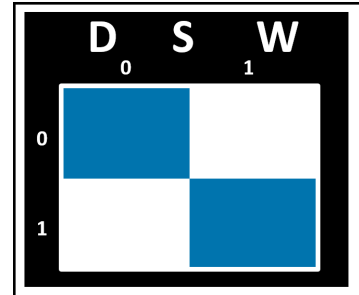
גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 2 מתוך 8



## הקדמה:

בעקבות הצלחתה הרבה של מערכת ניהול העסקאות, חברת המכונות "DSW" מעוניינת לבנות מערכת נוספת שתתמוך בניהול המכירות של סוכנויות רכב. המערכת החדשה תנהל מידע רק ברמת סוגי הרכב (כלומר אין התייחסות לדגמים). סוכנויות רכב יכולות להתאחד, כאשר שתי סוכנויות רכב מתאחדות המכירות שלהן מתאחדות גם כן.



## דרוש מבנה נתונים למימוש הפעולות הבאות:

`void * Init()`

מאתחל מבנה נתונים ריק.

פרמטרים: אין.

ערך החזרה: מצביע למבנה נתונים ריק או NULL במקרה של כישלון.

סיבוכיות זמן:  $O(1)$  במקרה הגרוע.

`StatusType AddAgency(void *DS)`

הוספת סוכנות רכב למערכת. המספר המזהה של הסוכנות הראשונה המתווספת הוא 0, וכל סוכנות תקבל את המספר הטבעי הבא. כלומר, אם בעת הקריאה לפעולה יש  $n$  סוכנויות רכב קיימות, המספר המזהה של הסוכנות החדשה הוא  $n$ .

פרמטרים: DS מצביע למבנה הנתונים.

ערך החזרה: ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID\_INPUT אם DS=NULL.

SUCCESS במקרה של הצלחה.

סיבוכיות זמן:  $O(1)$  משוערך לבד.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 3 מתוך 8



**StatusType** SellCar(void \*DS, int agencyID, int typeId, int k)

עדכון מבנה הנתונים במכירת  $k$  רכבים מסוג typeId על ידי הסוכנות בעלת המזהה agencyID.

**פרמטרים:** DS מצביע למבנה הנתונים.

agencyID מזהה סוכנות הרכב.

typeID מזהה סוג הרכב.

k מספר הרכבים שנמכרו.

**ערך החזרה:** ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID\_INPUT אם  $DS == \text{NULL}$ ,  $k \leq 0$  או  $\text{agencyID} < 0$ .

FAILURE אם לא קיימת סוכנות רכב עם המזהה הנתון.

SUCCESS במקרה של הצלחה.

**סיבוכיות זמן:**  $O(\log^*(n) + \log(m))$  משוערך יחד עם UniteAgencies ו-GetlthSoldType. כאשר  $n$  הוא

מספר סוכנויות הרכב במערכת ו- $m$  הוא מספר סוגי רכב בסוכנות agencyID.

כל ערך מספרי הוא חוקי עבור typeId (גם מספרים שליליים)

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

הסוכנויות agencyID1 ו-agencyID2 מתאחדות. כל קריאה עתידית ל-agencyID1 או ל-agencyID2 תתייחס

לסוכנות המאוחדת. המכירות של הסוכנות המאוחדת הן סך המכירות של שתי הסוכנויות שאוחדו.

ניתן להניח שבשתי הסוכנויות לא נמצאים סוגי רכב זהים (כלומר typeId הוא מזהה ייחודי).

ניתן להניח שלא יהיה ניסיון לאחד סוכנות קיימת עם עצמה (לא ישירות ולא בעקיפין דרך סוכנויות שאוחדו איתה).

**פרמטרים:** DS מצביע למבנה הנתונים.

agencyID1/2 מזהי סוכנויות הרכב.

**ערך החזרה:** ALLOCATION\_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID\_INPUT אם  $DS == \text{NULL}$  או  $\text{agencyID1/2} < 0$ .

FAILURE אם לא קיימת סוכנות רכב עם המזהה הנתון.

SUCCESS במקרה של הצלחה.

**סיבוכיות זמן:**  $O(\log^*(n) + m_1 + m_2)$  משוערך יחד עם SellCar ו-GetlthSoldType. כאשר  $n$  הוא מספר

סוכנויות הרכב במערכת,  $m_1$  הוא מספר סוגי הרכב בסוכנות agencyID1 ו- $m_2$  הוא מספר סוגי הרכב בסוכנות

agencyID2.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 4 מתוך 8



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

הפעולה מחזירה את סוג הרכב שנמכר במקום ה- $i$  מבין כלל סוגי הרכב שנמכרו בסוכנות `agencyID`. כלומר, אם  $i$  שווה 0 הפעולה תחזיר את סוג הרכב שנמכר הכי מעט בסוכנות (בלי לקחת בחשבון סוגי רכב שלא נמכרו בסוכנות). במקרה של שוויון במספר המכירות הדגם שהמזהה שלו נמוך יותר ימוקם נמוך יותר בטבלת המכירות.

פרמטרים: `DS` מצביע למבנה הנתונים.

`agencyID` מזהה סוכנות הרכב.

`i` המיקום של סוג הרכב בטבלת המכירות.

`res` מצביע למשתנה שיעודכן להכיל את התוצאה.

ערך החזרה: `ALLOCATION_ERROR` במקרה של בעיה בהקצאת זכרון.

`INVALID_INPUT` אם `agencyID < 0` או `DS == NULL`, `res == NULL`, `i < 0`.

`FAILURE` אם לא קיימת סוכנות רכב עם המזהה הנתון או שבסוכנות

הרכב לא נמכרו  $i+1$  רכבים.

`SUCCESS` במקרה של הצלחה.

סיבוכיות זמן:  $O(\log^*(n) + \log(m))$  משוערך יחד עם `SellCar` ו-`UniteAgencies`. כאשר  $n$  הוא מספר

סוכנויות הרכב במערכת ו- $m$  הוא מספר סוגי הרכב בסוכנות `agencyID`.

`void Quit(void **DS)`

הפעולה משחררת את המבנה. בסוף השחרור יש להציב ערך `NULL` ב-`DS`, אף פעולה לא תקרא לאחר מכן

פרמטרים: `DS` מצביע למבנה הנתונים.

ערך החזרה: אין.

$O(n + m)$  כאשר  $n$  הוא מספר סוכנויות הרכב במערכת ו- $m$  הוא מספר סוגי הרכב הכולל בכל סוכנויות הרכב.

סיבוכיות מקום (עבור המבנה וכל הפעולות) -

$O(n + m)$  כאשר  $n$  הוא מספר סוכנויות הרכב במערכת ו- $m$  הוא מספר סוגי הרכב הכולל בכל סוכנויות הרכב.

## ערכי החזרה של הפונקציות:

בכל אחת מהפונקציות, ערך ההחזרה שיוחזר ייקבע לפי הכלל הבא:

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 5 מתוך 8



- תחילה, יוחזר INVALID\_INPUT אם הקלט אינו תקין.
  - אם לא הוחזר INVALID\_INPUT:
    - בכל שלב בפונקציה, אם קרתה שגיאת הקצאה יש להחזיר ALLOCATION\_ERROR.
    - אם קרתה שגיאה אחרת, כפי שמצוין בכל פונקציה, יש להחזיר מיד FAILURE מבלי לשנות את מבנה הנתונים.
  - אחרת יוחזר SUCCESS.
- מצביעים הנשלחים לפונקציות על מנת לקבל ערך החזרה לא נקראים כלל בבדיקה האוטומטית אם הוחזר מהפונקציה ערך ששונה מ-SUCCESS ולכן אין חשיבות לערך המוחזר בהם במקרים האלו.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 6 מתוך 8



## הנחיות:

### חלק יבש:

- **הציון על החלק היבש הוא 50% מהציון של התרגיל.**
- לפני מימוש הפעולות בקוד יש לתכנן היטב את מבני הנתונים והאלגוריתמים ולוודא כי באפשרותכם לממש את הפעולות בדרישות הזמן והזיכרון שלעיל.
- הגשת החלק הרטוב מהווה תנאי הכרחי לקבלת ציון על החלק היבש, כלומר, הגשה בה יתקבל אך ורק חלק יבש תגרוור ציון 0 על התרגיל כולו.
- יש להכין מסמך הכולל תיאור של מבני הנתונים והאלגוריתמים בהם השתמשתם בצירוף הוכחת סיבוכיות הזמן והמקום שלהם. חלק זה עומד בפני עצמו וצריך להיות מובן לקורא גם לפני העיון בקוד. אין צורך לתאר את הקוד ברמת המשתנים, הפונקציות והמחלקות, אלא ברמה העקרונית.
- ראשית הציגו את מבני הנתונים בהם השתמשתם. רצוי ומומלץ להיעזר בציור.
- לאחר מכן הסבירו כיצד מימשתם כל אחת מהפעולות הנדרשות. הוכיחו את דרישות סיבוכיות הזמן של כל פעולה תוך כדי התייחסות לשינויים שהפעולות גורמות במבני הנתונים.
- הוכיחו שמבנה הנתונים וכל הפעולות עומדים בדרישת סיבוכיות המקום.
- החסמים הנתונים בתרגיל הם לא בהכרח הדוקים ולכן יכול להיות שקיים פתרון בסיבוכיות טובה יותר. מספיק להוכיח את החסמים הדרושים בתרגיל.
- רמת פירוט: יש להסביר את כל הפרטים שאינם טריוויאליים ושחשובים לצורך מימוש הפעולות ועמידה בדרישות הסיבוכיות. אין לדון בפרטים טריוויאליים (הפעילו את שיקול דעתכם בקשר לזה, ושאלו את האחראי על התרגיל אם אינכם בטוחים). אין לצטט קטעים מהקוד כתחליף להסבר. אין צורך לפרט אלגוריתמים שנלמדו בכתה. כמו כן, אין צורך להוכיח תוצאות ידועות שנלמדו בכתה, אלא מספיק לציין בבירור לאיזו תוצאה אתם מתכוונים.
- **על חלק זה לא לחרוג מ-8 עמודים.**
- והכי חשוב **!keep it simple**

### חלק רטוב:

- מומלץ לממש תחילה את מבני הנתונים בצורה הכללית ביותר ורק אז לממש את הפונקציות הנדרשות בתרגיל.
- **אנו ממליצים בחום על מימוש Object Oriented, ++C, ++C++**, מימוש כזה יאפשר לכם להגיע לפתרון פשוט וקצר יותר לפונקציות אותן עליכם לממש ויאפשר לכם להכליל בקלות את מבני הנתונים שלכם (זכרו שיש תרגיל רטוב נוסף בהמשך הסמסטר). על מנת לעשות זאת הגדירו מחלקה, למשל בשם `CarDealershipManager`, וממשו בה את דרישות התרגיל. אח"כ, על מנת לייצר התאמה לממשק ה `C` ב `library2.h`, ממשו את `library.cpp` באופן הבא:

```
#include "library2.h"
#include "CarDealershipManager.h"

void *Init() {
    CarDealershipManager *DS = new CarDealershipManager ();
    return (void*)DS;
}

StatusType AddCarType(void *DS, int typeId, int numOfModels) {
    return ((CarDealershipManager *)DS)-> AddCarType(typeId, numOfModels);
}
```

- על הקוד להתקמפל על `cs13` באופן הבא:

**g++ -std=c++11 -DNDEBUG -Wall \*.cpp**

עליכם מוטלת האחריות לוודא קומפילציה של התכנית ב- `g++`. אם בחרתם לעבוד בקומפיילר אחר, מומלץ לקמפל ב- `g++` מידי פעם במהלך העבודה.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 7 מתוך 8



- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ [library2.h](#).
- קראו היטב את הקובץ הנ"ל, לפני תחילת העבודה.
- אין לשנות את הקבצים אשר סופקו כחלק מהתרגיל, ואין להגיש אותם.
- עליכם לממש בעצמכם את כל מבני הנתונים (למשל אין להשתמש במבנים של STL ואין להוריד מבני נתונים מהאינטרנט). **כחלק מתהליך הבדיקה אנו נבצע בדיקה ידנית של הקוד ונוודא שאכן מימשתם את מבני הנתונים שבהם השתמשתם.**
- יש לתעד את הקוד בצורה נאותה וסבירה.
- מצורפים לתרגיל קבצי קלט ופלט לדוגמה.
- שימו לב: התוכנית שלכם תיבדק על קלטים שונים מקבצי הדוגמה הנ"ל, שיהיו ארוכים ויכללו מקרי קצה שונים. לכן, מומלץ מאוד לייצר בעצמכם קבצי קלט, לבדוק את התוכנית עליהם, ולוודא שהיא מטפלת נכון בכל מקרה הקצה.

## הגשה:

- חלק יבש + חלק רטוב:
  - הגשת התרגיל הנה **אך ורק** אלקטרונית דרך אתר הקורס.
  - יש להגיש קובץ ZIP שמכיל את הדברים הבאים:
    - בתיקייה הראשית:
      - קבצי ה-Source Files שלכם (ללא הקבצים שפורסמו).
      - קובץ PDF אשר מכיל את הפתרון היבש עבור. מומלץ להקליד את החלק הזה אך ניתן להגיש קובץ PDF מבוסס על סריקה של פתרון כתוב בכתב יד. שימו לב כי במקרה של כתב לא קריא, כל החלק השני לא תיבדק.
      - קובץ submissions.txt, המכיל בשורה הראשונה את שם, תעודת הזהות וכתובת הדוא"ל של השותף הראשון ובשורה השנייה את שם, תעודת הזהות וכתובת הדוא"ל של השותף השני. לדוגמה:

Roi Bar Zur 012345678 [idoغالil@cs.technion.ac.il](mailto:idoغالil@cs.technion.ac.il)

Henry Taub 123456789 [taub@cs.technion.ac.il](mailto:taub@cs.technion.ac.il)

- שימו לב כי אתם מגישים את כל שלושת החלקים הנ"ל.
- אין להשתמש בפורמט כיווץ אחר (לדוגמה RAR), מאחר ומערך הבדיקה האוטומטי אינו יודע לזהות פורמטים אחרים.
- יש לוודא שכאשר נכנסים לקובץ הזיפ הקבצים מופיעים מיד בתוכו ולא בתוך תיקיה שבתוך קובץ הזיפ. עבור הגשה שבה הקבצים יהיו בתוך תיקייה, הבדיקה האוטומטית לא תמצא את הקבצים ולא תוכל לקמפל ולהריץ את הקוד שלכם ולכן תיתן אוטומטית 0.
- לאחר שהגשתם, יש באפשרותכם לשנות את התוכנית ולהגיש שוב.
- ההגשה האחרונה היא הנחשבת.
- הגשה שלא תעמוד בקריטריונים הנ"ל תפסל ותקנס בנקודות!

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 27.05.2021

עמוד 8 מתוך 8



## דחיות ואיחורים בהגשה:

- דחיות בתרגיל הבית תינתנה אך ורק לפי תקנון הקורס.
- 5 נקודות יורדו על כל יום איחור בהגשה ללא אישור מראש. באפשרותכם להגיש תרגיל באיחור של עד 5 ימים ללא אישור. תרגיל שיוגש באיחור של יותר מ-5 ימים ללא אישור מראש יקבל 0.
- במקרה של איחור בהגשת התרגיל יש עדיין להגיש את התרגיל אלקטרונית דרך אתר הקורס.
- בקשות להגשה מאוחרת יש להפנות באמצעות [הטופס](#) האינטרנטי. לאחר קבלת אישור במייל על הבקשה, מספר הימים שאושרו לכם נשמר אצלנו. לכן, אין צורך לצרף להגשת התרגיל אישורים נוספים או את שער ההגשה באיחור.

**בהצלחה!**