

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 1 מתוך 9



מרח גומייד, [marahdss21@gmail.com](mailto:marahdss21@gmail.com)

**מתרגל ממונה על התרגיל:**

18/05/2021 בשעה 23:55

**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 בפיאצה 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

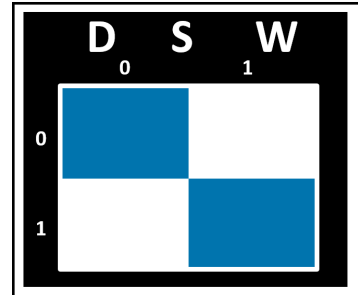
גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 2 מתוך 9



## הקדמה:

חברת המכוניות הענקית "DSW" מעוניינת לפתח מערכת שתקל על ניהול העסקאות שתאפשר לחברה וללקוחות שלה לבצע את העסקאות בנוחות. החברה מייצרת מספר גדול של סוגי מכוניות וכל סוג מחולק גם לדגמים, לכל דגם החברה נותנת מספר זיהוי ומחזיקה מכונית אחת מדגם זה להצגה. כשלקוח רוצה לקנות מכונית מסויימת, הוא מציין את הסוג ואת הדגם של המכונית שהוא רוצה לקנות. המערכת תתמוך בבקשות של לקוחות לקנות מכוניות, להגיש תלונות, ותאפשר לחברה לעקוב אחרי מספרי המכירות מכל דגם והתלונות על כל דגם על מנת לדעת איזה דגמים הם הכי פופולריים במטרה לשיפור האיכות של המכוניות שהיא מייצרת.



לכל דגם מחושב ציון באופן הבא: הציון מתחיל מ-0, עבור כל מכירה המכונית מקבלת 10 נקודות, ועבור כל תלונה המכונית מפסידה  $\lfloor 100/t \rfloor$  (100 חלקי t מעוגל למטה) נקודות כאשר t הוא מספר החודשים שעברו מאז שהלקוח קנה את המכונית.

## 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 AddCarType (void *DS, int typeId, int numOfModels)`

הוספת סוג רכב חדש עם מזהה typeId עם numOfModels דגמים, המספרים של הדגמים של סוג המכונית שהתווספה הם מספרים עוקבים החל מ-0 (0,1,2,... וכן הלאה). כל דגם מאותחל עם 0 מכירות ו-0 תלונות.

פרמטרים: DS מצביע למבנה הנתונים.

typeId מזהה סוג הרכב.

numOfModels מספר הדגמים של סוג הרכב.

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

INVALID\_INPUT אם DS=NULL, numOfModels<=0 או typeId<=0

FAILURE אם כבר קיים סוג רכב עם המספר המזהה הנתון.

SUCCESS במקרה של הצלחה.

סיבוכיות זמן:  $O(\log(n) + m)$  במקרה הגרוע, כאשר n הוא מספר הדגמים במערכת בזמן הפעולה ו-m הוא numOfModels.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 3 מתוך 9



**StatusType RemoveCarType**(void \*DS, int typeId)

מחיקת סוג הרכב עם המזהה typeId וכל הדגמים של רכב זה.

**פרמטרים:** DS מצביע למבנה הנתונים.

typeID מזהה סוג הרכב.

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

INVALID\_INPUT אם DS==NULL או typeId<=0.

FAILURE אם לא קיים סוג רכב עם המספר המזהה הנתון.

SUCCESS במקרה של הצלחה.

**סיבוכיות זמן:**  $O(\log(n) + m \log(M))$  במקרה הגרוע, כאשר n הוא מספר סוגי הרכבים במערכת בזמן

הפעולה, M הוא מספר הדגמים הכולל במערכת, ו-m הוא מספר הדגמים של הרכב שהסרנו.

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

מכירה של המכונית carID מהדגם modelID.

**פרמטרים:** DS מצביע למבנה הנתונים.

typeID מזהה סוג הרכב.

modelID מזהה הדגם.

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

INVALID\_INPUT אם DS==NULL, typeId<=0, modelID<0 או

modelID>=numOfModels (כאשר numOfModels הוא מספר הדגמים של סוג הרכב ה"ל).

FAILURE אם לא קיים סוג רכב עם המזהה הנתון.

SUCCESS במקרה של הצלחה.

**סיבוכיות זמן:**  $O(\log(n) + \log(M))$  במקרה הגרוע, כאשר n הוא מספר סוגי הרכבים במערכת בזמן

הפעולה, M הוא מספר הדגמים הכולל במערכת.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 4 מתוך 9



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

הגשת תלונה עבור המכונית carID מהדגם modelID.

פרמטרים:	DS	מצביע למבנה הנתונים.
	typeID	מזהה סוג הרכב.
	modelID	מזהה הדגם.
	t	מספר החודשים מרגע קניית הרכב.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS=NULL, typeId<=0, modelID<0 או modelID>numOfModels (כאשר numOfModels הוא מספר הדגמים של סוג הרכב ה"ל").
	FAILURE	אם לא קיים דגם עם המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(M))$	במקרה הגרוע, כאשר M הוא מספר הדגמים הכולל במערכת בזמן הפעולה.

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

החזרת המזהה של המכונית בעלת מספר המכירות הגבוה ביותר מסוג הרכב typeId.

- אם typeId=0 יש להחזיר את הדגם בעלת מספר מכירות הכי גבוה בכל המערכת.
- אם לשני דגמים יש מספר מכירות מקסימלי, יש להחזיר את הדגם בעל מספר סוג הרכב הקטן מבין השניים, אם סוג הרכב זהה, יש להחזיר את הדגם בעל מזהה סוג הדגם הקטן מבין השניים.

פרמטרים:	DS	מצביע למבנה הנתונים.
	typeID	מזהה סוג הרכב.
	modelID	מצביע למשתנה שיעודכן למזהה של הדגם בעל מספר המכירות הכי גבוה. אין צורך להחזיר את סוג הרכב.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS=NULL, typeId<0 או typeId<0.
	FAILURE	אם לא קיים סוג רכב עם המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(1)$	אם typeId=0 אז במקרה הגרוע. אחרת, $O(\log(n))$ במקרה הגרוע, כאשר n הוא מספר סוגי הרכב במערכת בזמן הפעולה.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 5 מתוך 9



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

החזרת `numOfModels` הדגמים בעלי הציון הכי נמוך במערכת בסדר עולה. הציון מחושב לפי מספר המכירות ומספר התלונות עבור כל מכונית כפי שהוסבר בתחילת התרגיל. יש להחזיר עבור כל דגם את המספר המזהה של סוג הרכב שלה במערך `types` והמספר המזהה של הדגם במערך `models`, כך שהמקומות בשני המערכים תואמים זה לזה. כלומר, מזהה סוג הרכב של הדגם עם הציון ה-`i` הנמוך ביותר יהיה בתא מספר `i` במערך `types`, ומזהה הדגם שאליו היא שייכת יהיה בתא `i` במערך `models`. במקרה של שוויון בציון, יש להחזיר את הדגם בעל מזהה סוג הרכב הקטן מבין השניים, אם סוג הרכב זהה, יש להחזיר את הדגם בעל מזהה סוג הדגם הקטן מבין השניים.

פרמטרים: `DS` מצביע למבנה הנתונים.

`numOfCars` כמות הדגמים שיש להחזיר.

`types` מערך שיכיל את מזהי סוגי הרכב.

`models` מערך שיכיל את מזהי הדגמים.

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

`INVALID_INPUT` אם `DS==NULL` או `numOfModels<=0`.

`FAILURE` אם יש במבנה פחות מ-`numOfModels` דגמים.

`SUCCESS` במקרה של הצלחה.

סיבוכיות זמן:  $O(m)$  במקרה הגרוע, כאשר `m` הוא מספר הדגמים המבוקשים (`numOfModels`).

שימו לב שהמערכים כבר מוקצים בגודל המתאים, יש רק למלא אותם.

# מבני נתונים 234218 אביב תשפ"א

גיליון רטוב מספר 1 – מעודכן לתאריך 29.04.2021

עמוד 6 מתוך 9



`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.

**סיבוכיות זמן:**  $O(n + m)$  במקרה הגרוע, כאשר n הוא מספר הדגמים ו-m הוא מספר המכונות הכולל מכל הדגמים.

**סיבוכיות מקום (עבור המבנה וכל הפעולות):**  $O(n + m)$  במקרה הגרוע, כאשר n הוא מספר הדגמים ו-m הוא מספר המכונות הכולל מכל הדגמים.

## ערכי החזרה של הפונקציות:

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

- 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 1 – מעודכן לתאריך 29.04.2021

עמוד 7 מתוך 9



## 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 library.h, ממשו את library.cpp באופן הבא:

```
#include "library.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:

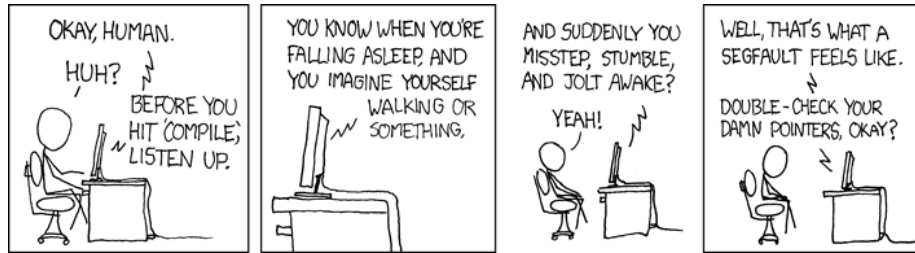
**g++ -std=c++11 -DNDEBUG -Wall \*.cpp**

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 1 – מעודכן לתאריך 29.04.2021

עמוד 8 מתוך 9



- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ `library.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 קלט 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 1 – מעודכן לתאריך 29.04.2021

עמוד 9 מתוך 9



## 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!**