

# מבני נתונים 234218 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 1 מתוך 8



23:55 בשעה 17/01/2020

**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.**
- שאלות על התרגיל יש לפרסם באתר [הפיאצה של הקורס](#).
- בקשות להגשה מאוחרת יש להפנות באמצעות [הטופס](#) האינטרנטי.

# מבני נתונים 234218 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 2 מתוך 8



## הקדמה:

חברת "boom" המנהלת פלטפורמת פגישות מקוונות החליטה לשפר את המערכת שלה לצפייה בשיעורים מוקלטים בעקבות ההצלחה של הגרסה הראשונה. כמו קודם, החברה רוצה מערכת שתקל על יצירת קורסים עם שיעורים מוקלטים רבים, שתאפשר לסטודנטים ולתלמידים הרשומים לאותם קורסים לצפות בהם בנוחות. השיפורים הנדרשים הם שיפורי סיבוכיות וגם הפעם יהיה ניתן להוסיף שיעורים לקורסים קיימים.

המערכת מחכה לבקשות של סטודנטים לצפות בשיעור של קורס מסוים למשך זמן רצוי, ומאפשרת למוסד האקדמי לעקוב אחרי השיעורים הכי נצפים מבחינת משך זמן צפייה ומספר צפיות כולל בשביל לאפשר את שיפור איכות ההוראה ע"י



מדידת התמדת הסטודנטים.

## 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 AddCourse(void* DS, int courseId)`

הוספת קורס חדש עם מזהה `courseID`. הוא מאותחל להכיל 0 הרצאות.

**פרמטרים:** `DS` מצביע למבנה הנתונים.

`courseID` מזהה קורס.

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

`INVALID_INPUT` אם `DS==NULL` או `courseID<=0`

`FAILURE` אם כבר קיים קורס עם המספר המזהה הנתון.

`SUCCESS` במקרה של הצלחה.

**סיבוכיות זמן:**  $O(1)$  בממוצע על הקלט, משוערך.

`StatusType RemoveCourse(void* DS, int courseId)`

מחיקת קורס עם המזהה `courseID` וכל השיעורים המועברים בו.

**פרמטרים:** `DS` מצביע למבנה הנתונים.

`courseID` מזהה קורס.

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

`INVALID_INPUT` אם `DS==NULL` או `courseID<=0`

`FAILURE` אם לא קיים קורס עם המספר המזהה הנתון.

`SUCCESS` במקרה של הצלחה.

# מבני נתונים 234218 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 3 מתוך 8



**סיבוכיות זמן:**  $O(m \log(M))$  בממוצע על הקלט, משוערך, כאשר  $M$  הוא מספר ההרצאות במערכת בזמן הפעולה ו- $m$  הוא מספר ההרצאות של הקורס שהסרנו ( $m$  הוא 1 אם אין קורס עם המזהה הנתון).

**StatusType AddClass(void\* DS, int courseID, int\* classID)**

הוספת שיעור חדש לקורס קיים עם מזהה courseID. השיעור יקבל את המזהה הכי קטן מהקבוצה  $\{0, 1, 2, \dots\}$  שעדיין לא הוקצה לקורס הנתון.

<b>פרמטרים:</b>	DS	מצביע למבנה הנתונים.
	courseID	מזהה קורס.
	classID	מצביע למשתנה שמכיל את מזהה השיעור שהתווסף.
<b>ערך החזרה:</b>	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == \text{NULL}$ או $courseID \leq 0$
	FAILURE	אם לא קיים קורס עם המספר המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
<b>סיבוכיות זמן:</b>	$O(1)$	בממוצע על הקלט, משוערך.

**StatusType WatchClass(void\* DS, int courseID, int classID, int time)**

סימון שבוצעה צפייה של שיעור classID של קורס courseID במשך time דקות.

<b>פרמטרים:</b>	DS	מצביע למבנה הנתונים.
	courseID	מזהה קורס.
	classID	מזהה השיעור.
	time	משך זמן הצפייה (בדקות).
<b>ערך החזרה:</b>	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == \text{NULL}$ , $courseID \leq 0$ , $classID < 0$ , $time \leq 0$ או $classID + 1 > \text{numOfClasses}$ (כאשר numOfClasses הוא מספר השיעורים בקורס).
	FAILURE	אם לא קיים קורס עם המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
<b>סיבוכיות זמן:</b>	$O(\log(M + 2))$	בממוצע על הקלט, כאשר $M$ הוא מספר ההרצאות במערכת בזמן הפעולה (שימו לב שה-2 מטרותו רק שהארגומנט של ה-log והערך שלו לא יתאפסו).

# מבני נתונים 234218 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 4 מתוך 8



**StatusType** TimeViewed(void\* DS, int courseID, int classID, int\* timeViewed)

החזרת משך הצפייה הכולל של שיעור classID של קורס courseID לתוך המשתנה timeViewed.

פרמטרים:	DS	מצביע למבנה הנתונים.
	courseID	מזהה קורס.
	classID	מזהה השיעור.
	timeViewed	מצביע למשתנה שיכיל את משך זמן הצפייה הכולל בשיעור.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS=NULL, courseID<=0, classID<0 או classID+1>numOfClasses (כאשר numOfClasses הוא מספר השיעורים בקורס).
	FAILURE	אם לא קיים קורס עם המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(1)$	בממוצע על הקלט.

**StatusType** GetIthWatchedClass(void\* DS, int i, int\* courseID, int\* classID)

החזרת השיעור ה-i בסדרת השיעורים הכי נצפים מבין כל השיעורים במערת – מבחינת סכום משך זמני הצפייה בדקות – בסדר יורד. במקרה של שוויון בסך דקות הצפייה, יש להעדיף את המספר המזהה של הקורס בסדר עולה, ובמקרה של שוויון נוסף יש להחזיר לפי מזהה השיעור בסדר עולה. הפעולה מעניינת רק עבור שיעורים שיש בהם צפייות, לכן אם יש במערכת פחות מ-rank שיעורים עם צפייות אז הפעולה אמורה להיכשל.

פרמטרים:	DS	מצביע למבנה הנתונים.
	i	אינדקס השיעור שיש להחזיר בסדרת השיעורים הכי נצפים.
	courseID	מצביע למזהה קורס המוחזר.
	classID	מצביע למזהה השיעור המוחזר.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם DS=NULL או $i \leq 0$ .
	FAILURE	אם יש במבנה פחות מ-i שיעורים עם צפייות.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(M + 2))$	במקרה הגרוע, כאשר M הוא מספר השיעורים במערכת בזמן ביצוע הפעולה.

**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 הקורסים ו-m הוא מספר השיעורים הכולל בכל הקורסים.

# מבני נתונים 234218 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 5 מתוך 8



סיבוכיות מקום (עבור המבנה וכל הפעולות) -  $O(n + m)$  במקרה הגרוע, כאשר  $n$  הוא מספר הקורסים ו- $m$  הוא מספר השיעורים הכולל בכל הקורסים.

## 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 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

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. על מנת לעשות זאת הגדירו מחלקה, למשל בשם CoursesManager, וממשו בה את דרישות התרגיל. אח"כ, על מנת לייצר התאמה לממשק ה C ב library.h, ממשו את library.cpp באופן הבא:

```
#include "library.h"
#include "CoursesManager.h"

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

StatusType AddCourse(void *DS, int courseID, int numOfClass) {
    return ((CoursesManager *)DS)-> AddCourse(courseID, numOfCourses);
}
```

- על הקוד להתקמפל על `cs/3` 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 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

עמוד 7 מתוך 8



- 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 **ארכיון** אלקטרוני דרך אתר הקורס.
  - יש להגיש קובץ **ZIP** שמכיל את הדברים הבאים:
    - בתיקיה 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 חורף תשפ"א

גיליון רטוב מספר 2 – מעודכן לתאריך 23.12.2020

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!**