

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

עמוד 1 מתוך 9



18/06/2020 בשעה 23:59

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 במרכז 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 חובה להפנות באמצעות **הטופס** האינטרנטי **מראש**.

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

עמוד 2 מתוך 9



הקדמה:

בעקבות ההצלחה המסחררת של המערכת החדשה של חברת "דיזל", החברה "ענן שמע" החליטה גם לפתח מערכת ממוחשבת למעקב אחר השירים שלה. על מנת להתחרות במערכת של החברה השנייה, עליה לפתח מערכת מהירה יותר. כמו המערכת הקודמת, גם הפעם המערכת צריכה לתמוך במספר בלתי מוגבל של אומנים שלכל אחד מהם מספר שירים ולזכור כמה פעמים כל שיר הושמע. בנוסף, כל אומן וכל שיר מיוצגים על ידי מספר מזהה שהוא שלם



וחיובי.

בשונה מהמערכת הקודמת, על המערכת החדשה לאפשר למשתמש לקבל את השיר הכי מושמע של כל אומן ביעילות וגם לקבל כל שיר במיקום מסוים במצעד הלהיטים ביעילות.

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 AddArtist(void *DS, int artistID)`

הוספת אומן חדש עם מזהה `artistID`.

פרמטרים: `DS` מצביע למבנה הנתונים.

`artistID` מזהה האומן.

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

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

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

`SUCCESS` במקרה של הצלחה.

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

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020
עמוד 3 מתוך 9



StatusType RemoveArtist(void *DS, int artistID)

מחיקת האומן עם המזהה $artistID$ אם אין לו שירים.

פרמטרים:	DS	מצביע למבנה הנתונים.
	artistID	מזהה האומן.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$ או $artistID \leq 0$.
	FAILURE	אם לא קיים אומן עם המספר המזהה הנתון או אם קיים אומן כזה ויש לו שירים.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(1)$	בממוצע על הקלט משוערך.

StatusType AddSong(void *DS, int artistID, int songID)

הוספת שיר חדש של האומן $artistID$ עם המזהה $songID$. השיר מתחיל עם 0 השמעות.

פרמטרים:	DS	מצביע למבנה הנתונים.
	artistID	מזהה האומן.
	songID	מזהה השיר של האומן.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהתקצאת זכרון.
	INVALID_INPUT	אם $artistID \leq 0$ או $DS == NULL$, $songID \leq 0$.
	FAILURE	אם לא קיים אומן עם המספר המזהה הנתון או שקיים אומן כזה אבל כבר קיים לו שיר עם המספר המזהה.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(n))$	בממוצע על הקלט, כאשר n הוא מספר השירים הכולל במערכת.

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

עמוד 4 מתוך 9



`StatusType RemoveSong(void *DS, int artistID, int songID)`

מחיקת השיר `songID` של האומן `artistID`.

פרמטרים:	DS	מצביע למבנה הנתונים.
	artistID	מזהה האומן.
	songID	מזהה השיר של האומן.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם <code>artistID <= 0</code> או <code>DS == NULL</code> , <code>songID <= 0</code>
	FAILURE	אם לא קיים אומן עם המספר המזהה הנתון או שלא קיים לאמן שיר עם המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(n))$	בממוצע על הקלט, כאשר <code>n</code> הוא מספר השירים הכולל במערכת.

`StatusType AddToSongCount(void *DS, int artistID, int songID, int count)`

סימון שהתבצעו `count` השמעות נוספות של השיר `songID` של האומן `artistID`.

פרמטרים:	DS	מצביע למבנה הנתונים.
	artistID	מזהה האומן.
	songID	מזהה השיר של האומן.
	count	מספר ההשמעות החדשות.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם <code>artistID <= 0</code> או <code>DS == NULL</code> , <code>songID <= 0</code> , <code>count <= 0</code>
	FAILURE	אם לא קיים אומן עם המספר המזהה הנתון.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(n))$	בממוצע על הקלט, כאשר <code>n</code> הוא מספר השירים הכולל במערכת.

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

עמוד 5 מתוך 9



StatusType GetArtistBestSong(void *DS, int artistID, int *songID)

החזרת השיר הכי מושמע של האומן $artistID$. במקרה של שוויון יש את השיר עם המספר המזהה הקטן יותר.

פרמטרים:	DS	מצביע למבנה הנתונים.
	artistID	מזהה האומן.
	songID	מצביע למשתנה אשר בו שמים את המספר המזהה של השיר.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $artistID \leq 0$ או $DS == NULL$, $songID == NULL$.
	FAILURE	אם לא קיים אומן עם המספר המזהה הנתון או שקיים אומן כזה אבל אין לו שירים.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(1)$	בממוצע על הקלט.

StatusType GetRecommendedSongInPlace(void *DS, int rank, int *artistID, int *songID)

החזרת השיר במקום $rank$ במצעד הלהיטים. מצעד הלהיטים, כמו בתרגיל הקודם הוא השירים ממוינים לפי מספר השמעות בסדר יורד. במקרה של שוויון במספר ההשמעות, השוויון נשבר לפי המספר המזהה של האומן בסדר עולה ובמקרה של שוויון נוסף אז לפי מספר השיר בסדר עולה. יש להחזיר רק את השיר (המספר המזהה של האומן והמספר המזהה של השיר) במיקום הדרוש ולא את כל השירים עד המיקום הדרוש.

פרמטרים:	DS	מצביע למבנה הנתונים.
	rank	המיקום של השיר שיש להחזיר ממצעד הלהיטים.
	artistID	מצביע למשתנה אשר בו שמים את המספר המזהה של האומן.
	songID	מצביע למשתנה אשר בו שמים את המספר המזהה של השיר.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$, $rank \leq 0$, $artistID == NULL$ או $songID == NULL$.
	FAILURE	אם יש במבנה פחות מ- $rank$ שירים.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(n))$	במקרה הגרוע, כאשר n הוא מספר השירים הכולל במערכת.

מבני נתונים 234218 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

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 m הוא מספר האומנים במערכת ו-n הוא מספר השירים הכולל במערכת.

סיבוכיות מקום (עבור המבנה וכל הפעולות): $O(n + m)$ במקרה הגרוע, כאשר m הוא מספר האומנים במערכת ו-n הוא מספר השירים הכולל במערכת בכל שלב. (בכל זמן במהלך הריצה יש לקיים את החסם לפי המספר העדכני של האומנים והשירים במערכת באותו הרגע).

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 מיד. אין צורך לדאוג לתקינות המבנה.
 - 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 תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

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, 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 **MusicManager**, וממשו בה את דרישות התרגיל. אח"כ, על מנת לייצר התאמה לממשק ה C library2.h, ממשו את library2.cpp באופן הבא:

```
#include "library2.h"
#include "MusicManager.h"

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

StatusType AddArtist(void *DS, int artistID) {
    return ((MusicManager *)DS)-> AddArtist(artistID);
}
```

- 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 אביב תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

עמוד 8 מתוך 9



- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ `library1.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 לדוגמא.
- שימו לב: התוכנית שלכם תיבדק על קלטים שונים מקבצי הדוגמא 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:
 - 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 roi.bar-zur@cs.technion.ac.il

Henry Taub 123456789 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 הדיפ הקבצים מופיעים מיד בתוכו ולא בתוך תיקיה שבתוך קובץ הדיפ.** 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 תש"ף

גיליון רטוב מספר 2 – מעודכן לתאריך 02.06.2020

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!