

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

עמוד 1 מתוך 9



12/05/2019 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 בחוץ ה FAQ באתר הקורס 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 להפנות למייל: cs234218.technion@gmail.com
- בקשות להגשה מאוחרת יש להפנות באמצעות [הטופס](#) האינטרנטי.

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

עמוד 2 מתוך 9



הקדמה:

בעקבות השיפוצים בבניין טאוב, הפקולטה למדעי המחשב מעוניינת לפתח מערכת מחשוב חדשה שתאפשר לה לנהל את מערכת השעות של הקורסים בפקולטה. תפקיד המערכת יהיה להחזיק מידע על הקורסים, כיתות הלימוד והשיעורים המתקיימים בהם לפי השעה בה הם מתקיימים. השעות האפשריות במערכת מיוצגות על ידי מספרים סידוריים עוקבים. לדוגמא, השעה הראשונה במערכת היא 7:30-8:20 והיא מיוצגת על ידי 0, השעה השנייה היא 8:30-9:20 והיא מיוצגת על ידי 1 וכן הלאה. מספר השעות האפשריות במערכת ניתן מראש באתחול המערכת (נסמנו hours) ולאחר מכן נשאר קבוע. גם הכיתות במערכת מיוצגות על ידי מספרים סידוריים עוקבים שמתחילים ב-0 ומספרן ניתן באתחול המערכת (נסמנו rooms) ונשאר קבוע. כל קורס מיוצג על ידי מספר ייחודי חיובי (מספרים לא עוקבים). כל שיעור המיוצג במערכת מוגדר על ידי כיתה ושעה (זאת אומרת שבכיתה מסוימת ובשעה מסוימת יכול להתקיים שיעור אחד לכל היותר) וצריך לשמור את מספר הקורס שלו. שימו לב: לכל קורס יכולים להיות מספר שיעורים. בנוסף, לקורס יכולים להיות מספר שיעורים באותו הזמן (בחדרים שונים).



דרוש מבנה נתונים למימוש הפעולות הבאות:

`void * Init(int hours, int rooms)`

מאתחל מבנה נתונים ריק.

פרמטרים: hours מספר השעות במערכת.

rooms מספר החדרים במערכת.

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

סיבוכיות זמן: $O(kr)$ במקרה הגרוע כאשר k הוא מספר השעות (hours) ו-r הוא מספר החדרים (rooms).

`StatusType AddLecture(void *DS, int hour, int roomID, int courseID)`

הוספת הקורס courseID אם לא היה קיים כבר והוספת שיעור חדש בשעה hour בחדר roomID של הקורס courseID.

פרמטרים: DS מצביע למבנה הנתונים.

hour השעה בה מתקיים השיעור.

roomID הכיתה בה מתקיים השיעור.

courseID הקורס שהשיעור שייך לו.

ערך החזרה: ALLOCATION_ERROR במקרה של בעיה בהקצאת זכרון.

INVALID_INPUT אם $DS == NULL$, $hour < 0$, $roomID < 0$ או $courseID \leq 0$.

FAILURE אם כבר קיים שיעור בכיתה והשעה הנתונים.

SUCCESS במקרה של הצלחה.

סיבוכיות זמן: $O(\log(n) + \log(m))$ במקרה הגרוע, כאשר n הוא מספר הקורסים במערכת ו-m הוא מספר השיעורים של הקורס הנתון.

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019
עמוד 3 מתוך 9



StatusType GetCourseID(void *DS, int hour, int roomID, int *courseID)

מציאת מספר הקורס של השיעור שמתקיים בשעה $hour$ בחדר $roomID$.

פרמטרים:	DS	מצביע למבנה הנתונים.
	hour	השעה בה מתקיים השיעור.
	roomID	הכיתה בה מתקיים השיעור.
	courseID	מצביע למשתנה שיעודכן למספר הקורס שהשיעור שייך לו.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$, $hour < 0$, $roomID < 0$ או $courseID == NULL$.
	FAILURE	אם לא קיים שיעור בכיתה והשעה הנתונים.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(1)$	במקרה הגרוע.

StatusType DeleteLecture(void *DS, int hour, int roomID)

מחיקת השיעור המתקיים בשעה $hour$ בחדר $roomID$ ומחיקת הקורס שאליו שייך אם לא נשארו שיעורים ששייכים לו.

פרמטרים:	DS	מצביע למבנה הנתונים.
	hour	השעה בה מתקיים השיעור.
	roomID	הכיתה בה מתקיים השיעור.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$, $hour < 0$ או $roomID < 0$.
	FAILURE	אם לא קיים שיעור בכיתה והשעה הנתונים.
	SUCCESS	במקרה של הצלחה.
סיבוכיות זמן:	$O(\log(n) + \log(m))$	במקרה הגרוע, כאשר n הוא מספר הקורסים במערכת ו- m הוא מספר השיעורים של הקורס של השיעור שמוחקים ($m=1$ אם לא קיים שיעור).

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

עמוד 4 מתוך 9



StatusType ChangeCourseID(void *DS, int oldCourseID, int newCourseID)

שינוי מספר הקורס של הקורס *oldCourseID* ל-*newCourseID*. כל השיעורים של הקורס יהיו שייכים למספר הקורס החדש.

אם הקורס *newCourseID* כבר קיים אז הקורסים מתאחדים לקורס אחד וכל השיעורים של שניהם שייכים לקורס החדש.

<u>פרמטרים:</u>	DS	מצביע למבנה הנתונים.
	oldCourseID	הקורס שרוצים להחליף לו מספר.
	newCourseID	מספר הקורס החדש של הקורס.
<u>ערך החזרה:</u>	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $oldCourseID \leq 0$ או $DS == NULL$, $newCourseID \leq 0$.
	FAILURE	אם הקורס <i>oldCourseID</i> לא קיים.
	SUCCESS	במקרה של הצלחה.
<u>סיבוכיות זמן:</u>	$O(\log(n) + m)$	במקרה הגרוע, כאשר n הוא מספר הקורסים במערכת ו- m הוא מספר השיעורים הכולל של שני הקורסים <i>oldCourseID</i> ו- <i>newCourseID</i> .

StatusType CalculateScheduleEfficiency(void *DS, float *efficiency)

החזרת מידת היעילות של המערכת שמחושב על פי: מספר השיעורים במערכת חלקי (מספר החדרים כפול מספר השעות שמתקיים בהן לפחות שיעור אחד)

<u>פרמטרים:</u>	DS	מצביע למבנה הנתונים.
	efficiency	מצביע למשתנה שיעודכן למידת היעילות.
<u>ערך החזרה:</u>	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$.
	FAILURE	אם לא קיימים שיעורים במערכת.
	SUCCESS	במקרה של הצלחה.
<u>סיבוכיות זמן:</u>	$O(1)$	במקרה הגרוע.

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019
עמוד 5 מתוך 9



StatusType GetAllFreeRoomsByHour(void *DS, int hour, int **rooms, int* numOfRooms)

מציאת כל החדרים שלא מתקיים בהם שיעור בשעה $hour$. אין חשיבות לסדר ההחזרה.

פרמטרים:	DS	מצביע למבנה הנתונים.
	hour	השעה שעבורה נרצה לקבל את המידע.
	rooms	מצביע למערך אשר יכיל את מספרי החדרים הפנויים.
	numOfRooms	מצביע למשתנה אשר יכיל את כמות החדרים הפנויים.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם $DS == NULL$, $hour < 0$, $rooms == NULL$ או $numOfRooms == NULL$.
	FAILURE	אם לא קיימים חדרים פנויים.
	SUCCESS	במקרה של הצלחה.

סיבוכיות זמן: $O(r)$ במקרה הגרוע, כאשר r הוא מספר החדרים הפנויים בשעה הנתונה.

שימו לב שאתם מקצים את המערך בגודל המתאים, כמו כן אתם צריכים להקצות את המערך בעצמכם באמצעות `malloc` (כי הוא ישוחרר בקוד שניתן לכם באמצעות `free`).

StatusType GetAllLecturesByCourse(void *DS, int courseID, int **hours, int **rooms, int* numOfLectures)

יש להחזיר את כל השיעורים של הקורס $courseID$.

יש להחזיר שני מערכים באורך שווה אשר מכילים בתא אחד במערך הראשון את השעה של השיעור ובתא האחרון האינדקס במערך השני את מספר החדר של השיעור.

- השעות והחדרים יוחזרו ממיונים לפי השעה בסדר עולה, אם יש שני שיעורים באותה השעה אז יש למיין אותם בסדר עולה לפי מספר החדר.

פרמטרים:	DS	מצביע למבנה הנתונים.
	courseID	מספר הקורס שעבורו נרצה לקבל את המידע.
	hours	מצביע למערך אשר יכיל את השעות של השיעורים של הקורס.
	rooms	מצביע למערך אשר יכיל את החדרים של השיעורים של הקורס.
	numOfLectures	מצביע למשתנה שיעודכן לאורך של המערכים המוחזרים.
ערך החזרה:	ALLOCATION_ERROR	במקרה של בעיה בהקצאת זכרון.
	INVALID_INPUT	אם אחד המצביעים שווה ל- $NULL$ או $courseID \leq 0$.
	FAILURE	אם הקורס $courseID$ לא קיים.
	SUCCESS	במקרה של הצלחה.

סיבוכיות זמן: $O(\log(n) + m)$ במקרה הגרוע, כאשר n הוא מספר הקורסים במערכת ו- m הוא מספר השיעורים של הקורס שעבורו נרצה לקבל את המידע ($m=1$ אם הקורס לא קיים).

שימו לב שאתם מקצים את המערכים בגודל המתאים, כמו כן אתם צריכים להקצות את המערכים בעצמכם באמצעות `malloc` (כי הם ישוחררו בקוד שניתן לכם באמצעות `free`).

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

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: $O(kr + n)$ במקרה הגרוע, כאשר n הוא מספר הקורסים, k הוא מספר השעות (hours) ו- r הוא מספר החדרים (rooms).

סיבוכיות מקום: $O(kr + n)$ במקרה הגרוע, כאשר n הוא מספר הקורסים, k הוא מספר השעות (hours) ו- r הוא מספר החדרים (rooms).

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 תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

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, נאמר Schedule, 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 "Schedule.h"
```

```
void *Init(int hours, int rooms) {
    Schedule *DS = new Schedule(hours, rooms);
    return (void*)DS;
}
StatusType AddLecture(void *DS, int hour, int roomID, int courseID) {
    return ((Schedule *)DS)-> AddLecture(hour, roomID, courseID);
}
```

- על הקוד להתקמפל על `cs/2` 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 תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

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 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.
- לאחר שהגשתם, יש באפשרותכם לשנות את התוכנית ולהגיש שוב.
- ההגשה האחרונה היא הנחשבת.
- הגשה שלא תעמוד בקריטריונים הנ"ל תפסל ותקנס בנקודות!

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.

מבני נתונים 234218 אביב תשע"ט

גיליון רטוב מספר 1 – מעודכן לתאריך 07.04.2019

עמוד 9 מתוך 9



- 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!