

# מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מעודכן לתאריך 26.12.2017



עמוד 1 מתוך 6

מתרגל ממונה על התרגיל: יאיר פלדמן, [yairf11@cs.technion.ac.il](mailto:yairf11@cs.technion.ac.il)

תאריך ושעת הגשה: 09/01/2018 בשעה 23:30

אופן ההגשה: בזוגות.

## הנחיות:

- שאלות לגבי דרישות התרגיל יש להפנות באימייל לכתובת הנ"ל.
- 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.
- בתרגיל זה אין הגבלה על מבני הנתונים בהם אתם יכולים להשתמש. מותר וגם מומלץ להשתמש במבנים שמישתם בתרגילים הקודמים, אם הם מתאימים לדרישות הסיבוכיות הנוכחיות.

הקדמה: לאחר שצברו ניסיון רב בקרבות, הגלדיאטורים התארגנו בקבוצות אימון ולחימה משותפות. כל גלדיאטור יכול להשתייך לקבוצת אימון אחת. כמו כן, לצורך הערכת הגלדיאטורים, לכל גלדיאטור יהיה ציון שייקבע לפי הצלחותיו בקרבות בהן השתתף.

## הפעולות שבחן מבנה הנתונים צריך לתמוך:

`void* init(int n, int* trainingGroupsIDs)`  
מאתחל מערכת חדשה בעלת  $n$  קבוצות אימון. לכל קבוצת אימון מזהה מייצג שהוא מספר שלם. המזהים של הקבוצות נתונים ב-`trainingGroupsIDs`, כמו כן ניתן להניח כי המזהים הנתונים שונים זה מזה.  
פרמטרים:  $n$  מספר קבוצות האימון.  
`trainingGroupsIDs` מצביע למערך אשר מחזיק מזהים עבור קבוצות האימון.  
ערך החזרה: מצביע למבנה נתונים ריק או NULL במקרה של כישלון ( $n < 2$ ), אם אחד המזהים הוא מספר שלילי, או כל כישלון אחר).  
סיבוכיות:  $O(n)$  במקרה הגרוע, כאשר  $n$  מספר קבוצות האימון.

`StatusType addTrainingGroup (void* DS, int trainingGroupID)`  
קבוצת אימון בעלת מספר מזהה `trainingGroupID` מתווספת למערכת.  
פרמטרים: `DS` מצביע למבנה הנתונים.  
`trainingGroupID` מספרה המזהה של קבוצת האימון אשר מתווספת למערכת.  
ערך החזרה: `ALLOCATION_ERROR` במקרה של בעיה בהקצאת זכרון.  
`INVALID_INPUT` אם `DS==NULL` או `trainingGroupID < 0`



## עמוד 2 מתוך 6

אם קבוצת האימון שמספרה המזהה trainingGroupID כבר  
 התווספה למערכת, או במקרה של כל בעיה אחרת. FAILURE  
 במקרה של הצלחה. SUCCESS  
 סיבוכיות:  $O(\log(n))$  בממוצע על הקלט משוערך, כאשר  $n$  מספר קבוצות האימון במערכת.

`StatusType addGladiator(void* DS, int gladiatorID, int score, int trainingGroup)`

גלדיאטור בעל מספר מזהה gladiatorID מתווסף למערכת, ומצטרף לקבוצה trainingGroup.

פרמטרים: DS מצביע למבנה הנתונים.

מספרו המזהה של הגלדיאטור אשר מתווסף למערכת. gladiatorID

הציון של הגלדיאטור. score

הקבוצה אליה הגלדיאטור מצטרף. trainingGroup

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

INVALID\_INPUT אם  $DS == NULL$ ,  $trainingGroup < 0$ ,

$gladiatorID < 0$ ,  $score < 0$ , או  $score > 100$ .

אם הגלדיאטור שמספרו המזהה gladiatorID כבר התווסף FAILURE

למערכת, או שקבוצת האימון בעלת המזהה trainingGroup

לא קיימת במערכת, או במקרה של כל בעיה אחרת.

במקרה של הצלחה. SUCCESS

סיבוכיות:  $O(\log(m))$  בממוצע על הקלט משוערך, כאשר  $m$  מספר הגלדיאטורים במערכת.

`StatusType trainingGroupFight (void* DS, int trainingGroup1, int trainingGroup2, int k1, int k2)`

שתי קבוצות האימון בעלות המזהים trainingGroup1, trainingGroup2 נלחמות, כאשר כל קבוצת אימון

מביאה את  $k1/k2$  הגלדיאטורים בעלי הציונים הגבוהים ביותר שלה בהתאמה. הקבוצה המנצחת זוהי הקבוצה

אשר סכום הציונים של הגלדיאטורים שלה הוא הגדול ביותר. במקרה של תיקו, הקבוצה בעלת המזהה הנמוך יותר

מנצחת. לאחר הקרב הקבוצה שניצחה כובשת את הקבוצה המפסידה, והקבוצה המפסידה לא יכולה להשתתף יותר

בקרבות. עם זאת, עדיין ניתן להוסיף גלדיאטורים לקבוצה הנכבשת והיא עדיין קיימת במערכת עם אותם

גלדיאטורים בדיוק כמו לפני הקרב.

פרמטרים: DS מצביע למבנה הנתונים.

המזהה של קבוצת האימון הראשונה. trainingGroup1

המזהה של קבוצת האימון השנייה. trainingGroup2

מספר הגלדיאטורים מקבוצת האימון הראשונה. k1

מספר הגלדיאטורים מקבוצת האימון השנייה. k2



## עמוד 3 מתוך 6

**ערך החזרה:** `ALLOCATION_ERROR` במקרה של בעיה בהקצאת זכרון.  
`INVALID_INPUT` אם `k1 <= 0, k2 <= 0, DS == NULL`  
או `trainingGroup1 / trainingGroup2 < 0`  
**FAILURE** אם קבוצת האימון שמספרה המזהה `trainingGroup1 / trainingGroup2` לא נמצאת במערכת או שהיא אינה יכולה להשתתף בקרב, או שלקבוצה `trainingGroup1 / trainingGroup2` יש פחות מ-`k1 / k2` גלדיאטורים (בהתאמה), אם `trainingGroup1 == trainingGroup2` או במקרה של כל בעיה אחרת.  
**SUCCESS** במקרה של הצלחה.  
**סיבוכיות:**  $O(\log(n) + \log(m))$  בממוצע על הקלט משוערך, כאשר  $m$  מספר הגלדיאטורים במערכת, ו- $n$  מספר קבוצות האימון במערכת.

`StatusType getMinTrainingGroup (void* DS, int* trainingGroup)`  
הפעולה מחזירה את המזהה הקטן ביותר של קבוצת אימון שעדיין לא נכבשה.  
**פרמטרים:** `DS` מצביע למבנה הנתונים.  
`trainingGroup` מצביע לכתובת בה ישמור המספר של קבוצת האימון המבוקשת.  
**ערך החזרה:** `INVALID_INPUT` אם `DS == NULL` או `trainingGroup == NULL`  
**FAILURE** במקרה של כל בעיה אחרת.  
**SUCCESS** במקרה של הצלחה.  
**סיבוכיות:**  $O(1)$  במקרה הגרוע.

`void quit (void** DS)`  
משחרר את מבנה הנתונים. בסוף השחרור יש להציב ערך `NULL` ב-`*DS`.  
**פרמטרים:** `DS` מצביע למבנה הנתונים.  
**ערך החזרה:** אין.  
**סיבוכיות:**  $O(m + n)$  במקרה הגרוע, כאשר  $n$  מספר קבוצות האימון במערכת ו- $m$  מספר הגלדיאטורים במערכת.

### סיבוכיות מקום

סיבוכיות המקום של מבנה הנתונים היא  $O(m + n)$ , כאשר  $m$  מספר קבוצות האימון במערכת ו- $m$  מספר הגלדיאטורים במערכת.



## עמוד 4 מתוך 6

### חלק יבש:

- הציון על החלק היבש הוא 50% מציון התרגיל.
- לפני מימוש הפעולות בקוד יש לתכנן היטב את מבני הנתונים והאלגוריתמים ולוודא כי באפשרותכם לממש את הפעולות בדרישות הזמן והזיכרון שלעיל.
- מומלץ לממש תחילה את מבני הנתונים בצורה הכללית ביותר ורק אז לממש את הפונקציות הנדרשות.
- 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.
- 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.
- הגשת החלק הרטוב מהווה תנאי הכרחי לקבלת ציון על החלק היבש, כלומר, הגשה בה יתקבל אך ורק חלק יבש תגרור ציון 0 על התרגיל כולו.
- על חלק זה לא לחרוג מ-8 עמודים.

### חלק רטוב:

- אנו ממליצים בחום על מימוש **Object Oriented**, **C++**. על מנת לעשות זאת הגדירו מחלקה, נאמר **Colosseum**, וממשו בה את דרישות התרגיל. אח"כ, על מנת לייצר התאמה לממשק ה **library2.h**, ממשו את **library2.cpp** באופן הבא:

```
#include "library2.h"
#include "Colosseum.h"

void* init(int n, int* trainingGroupsIDs) {
    Colosseum* DS = new Colosseum(n, trainingGroupsIDs);
    return (void*) DS;
}

StatusType getMinTrainingGroup(void* DS, int* trainingGroup) {
    return ((Colosseum*) DS) -> getMinTrainingGroup(trainingGroup);
}
```



- על הקוד להתקמפל על t2 באופן הבא:

**g++ -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.

## הערות נוספות:

- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ library2.h.
- קראו היטב את הקובץ הנ"ל, לפני תחילת העבודה.
- אין לשנות את הקבצים הנ"ל ואין להגיש אותם.
- עליכם לממש בעצמכם את כל מבני הנתונים (אין להשתמש במבנים של STL ואין להוריד מבני נתונים מהאינטרנט).
- יש לתעד את הקוד בצורה נאותה וסבירה.
- מסופקת לכם דוגמא של קובץ קלט (in.txt) וקובץ הפלט (out.txt) המתאים לו.
- 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 ארוכים, 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 (ללא תיקיות או תתי תיקיות בתוכו) שמכיל את הדברים הבאים:
  - קבצי ה-Source Files שלכם (ללא הקבצים שפורסמו).
  - קובץ PDF אשר מכיל את הפתרון היבש. מומלץ להקליד את החלק הזה. ניתן להגיש קובץ PDF מבוסס על סריקה של פתרון כתוב בכתב יד. שימו לב כי במקרה של כתב לא קריא, כל התרגיל לא ייבדק.
  - 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:

Yair Feldman 012345678 [yairf11@cs.technion.ac.il](mailto:yairf11@cs.technion.ac.il)

Lucius Commodus 123456789 [commodus@cs.technion.ac.il](mailto:commodus@cs.technion.ac.il)

- שימו לב כי אתם מגישים את כל שלושת החלקים הנ"ל.
- אין להשתמש בפורמט כיווץ אחר, מאחר ומערך הבדיקה האוטומטי אינו יודע לזהות פורמטים אחרים.
- אין להגיש קובץ המכיל תתי תיקיות.
- לאחר שהגשתם, יש באפשרותכם לשנות את התוכנית ולהגיש שוב.
- ההגשה האחרונה היא הנחשבת.

# מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מעודכן לתאריך 26.12.2017



עמוד 6 מתוך 6

- הגשה שלא תעמוד בקריטריונים הבאים תפסל ותיקנס בנקודות!

## העתקות טטופלנה בחומרה! (Buh dum tss)

### 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.
- במקרה של איחור מוצדק, יש לצרף לקובץ ה PDF שלכם את סיבות ההגשה באיחור, לפי הטופס המופיע באתר, כולל סריקות של אישורי מילואים או אישורי נבחן.

**בהצלחה!**