

מבני נתונים 234218 חורף תשפ"ג

גיליון רטוב מספר 2 – מעודכן לתאריך 29.05.2023
עמוד 1 מתוך 6



מתרגל ממונה על התרגיל:

עומר ירושלמי, omeryer@cs.technion.ac.il

תאריך ושעת הגשה:

17/06/2023 בשעה 23:55

אופן ההגשה:

בזוגות. אין להגיש ביחידים. (אלא באישור מתרגל אחראי של הקורס)

הנחיות כלליות:

- שאלות על התרגיל יש לפרסם באתר הפיאצה של הקורס תחת לשונית "wet_2":
 - האתר: piazza.com/technion.ac.il/winter2023/234218
 - נא לקרוא את השאלות של סטודנטים אחרים לפני שמפרסמים שאלה חדשה, למקרה שנשאלה כבר.
- נא לקרוא את המסמך "נהלי הקורס" באתר הקורס. בנוסף, נא לקרוא בעיון את כל ההנחיות בסוף מסמך זה.
- בפורום הפיאצה ינוהל FAQ ובמידת הצורך יועלו תיקונים כהודעות נעוצות (Pinned Notes). תיקונים אלו מחייבים.
- התרגיל מורכב משני חלקים: יבש ורטוב.
 - לאחר קריאת כלל הדרישות, מומלץ לתכנן תחילה את מבני הנתונים על נייר. דבר זה יכול לחסוך לכם זמן רב.
 - לפני שאתם ניגשים לקודד את פתרוןכם, ודאו כי יש לכם פתרון העומד בכל דרישות הסיבוכיות בתרגיל. תרגיל שאינו עומד בדרישות הסיבוכיות יחשב כפסול.
 - את הפתרון שלכם מומלץ לחלק למחלקות שונות שאפשר לממש (ולבדוק!) בהדרגתיות.
 - המלצות לפתרון התרגיל נמצאות באתר הקורס תחת: "Programming Tips Session".
- המלצות לתכנות במסמך זה אינן מחייבות, אך מומלץ להיעזר בהן.
- 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.
- בקשות להגשה מאוחרת יש להפנות למתרגל האחראי בלבד בכתובת:
turotovsally@campus.technion.ac.il



חברת תקליטים ידועה רוצה ליעל את האופן שבו היא עובדת. לשם כך היא צריכה עזרה במימוש של מבנה נתונים שיתאים לצרכיה.

מטרת המערכת היא לאסוף מידע אודות הלקוחות והתקליטים שבחנות. לכל לקוח שנכנס לחנות יש מזהה ייחודי שיסומן על ידי c_id (לא ייתכן שני לקוחות בעלי אותו c_id) ומספר טלפון. בנוסף, כל לקוח יכול להיות חבר מועדון במידה וירצה בכך. התקליטים שבחנות יאופיינו על ידי r_id אשר יהיו ממוספרים באופן רציף מ 0 ועד למספר התקליטים פחות 1, ובכל תחילת חודש מספר התקליטים ישתנה בהתאם לחודש הנוכחי (ייתכן כי r_id של תקליט ישתנה בין חודשים) ובנוסף על ידי מספר העותקים ומספר הרכישות שבוצעו עבור תקליט זה. כל לקוח שהינו חבר מועדון אינו משלם ישירות על כל תקליט אלא רק בסוף כל חודש. על מנת לאפשר זאת על המערכת לשמור את סך הרכישות שביצע כל חבר מועדון. בנוסף, על מנת לעודד לקוחות להיות חברי מועדון, מידי פעם מתבצעות הגרלות וחלוקת פרסים לחברי המועדון אשר מזוילות את המחיר עליהם לשלם.

לשם כך, עליכם לממש מבנה נתונים התומך בפעולות הבאות:
(לאורך כל השאלה n הינו מספר הלקוחות במערכת בעוד ש m הינו מספר התקליטים).

`RecordsCompany()`

פונקציה בונה של המבנה. על הפונקציה לאתחל את מבנה הנתונים באופן ריק.
פרמטרים: אין.
ערך החזרה: אין.
סיבוכיות זמן: $O(1)$

`~RecordsCompany()`

פונקציה הורסת של המבנה. על הפונקציה לשחרר ולמחוק את מבנה הנתונים.
פרמטרים: אין.
ערך החזרה: אין.
סיבוכיות זמן: $O(n + m)$

`StatusType newMonth(int *records_stocks, int number_of_records)`

הפעולה מאפסת את סידור התקליטים ואת סכום הכסף שכל אדם חייב.
ניתן להניח כי פונקציה זו תקרא לפני כל פעולה הכוללת פעולה עם תקליטים.
פרמטרים: $records_stocks$ – מערך של מספרים המכיל בכל אינדקס את מספר העותקים של תקליט מסוים.
 $number_of_records$ – מספר התקליטים בחודש זה (תואם למספר התאים ב $records_stocks$).
ערך החזרה:
 $number_of_records < 0$ – `INVALID_INPUT` (ניתן להניח כי $records_stocks$ תקין).
`ALLOCATION_ERROR` – במידה והייתה שגיאת זיכרון.
`SUCCESS` – במידה והפעולה הצליחה.
סיבוכיות זמן: $O(n + m)$

`StatusType addCostumer(int c_id, int phone)`

הפונקציה מוסיפה לקוח חדש למבנה הנתונים.
פרמטרים: c_id – המזהה של הלקוח, $phone$ – מספר הטלפון שלו.
ערך החזרה:
 $c_id < 0$ או $phone < 0$ – `INVALID_INPUT`
`ALREADY_EXISTS` – אם c_id כבר קיים במערכת.
`ALLOCATION_ERROR` – במידה והייתה שגיאת זיכרון.
`SUCCESS` – במידה והפעולה הצליחה.
סיבוכיות זמן: $O(1)$ משוערך במוצע על הקלט.

`Output_t<int> getPhone(int c_id)`

הפונקציה מחזירה את מספר הטלפון של הלקוח.



פרמטרים: c_id – המזהה של הלקוח.
ערך החזרה:

$c_id < 0$ – *INVALID_INPUT*

c_id – *DOESNT_EXISTS* – אם לא קיים לקוח בעל c_id .

במידה והפעולה הצליחה מחזירה את מספר הטלפון.

סיבוכיות זמן: $O(1)$ בממוצע על הקלט.

StatusType makeMember(int c_id)

הפונקציה הופכת את הלקוח בעל ה c_id לחבר מועדון.

פרמטרים: c_id – המזהה של הלקוח.

ערך החזרה:

$c_id < 0$ – *INVALID_INPUT*

c_id – *DOESNT_EXISTS* – אם לא קיים לקוח בעל c_id .

c_id – *ALREADY_EXISTS* – אם הלקוח בעל c_id הוא כבר חבר מועדון.

c_id – *ALLOCATION_ERROR* – במידה והייתה שגיאת זיכרון.

c_id – *SUCCESS* – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(\log n)$.

Output_t<bool> isMember(int c_id)

הפונקציה מחזירה האם הלקוח בעל ה c_id הינו חבר מועדון.

פרמטרים: c_id – המזהה של הלקוח.

ערך החזרה:

$c_id < 0$ – *INVALID_INPUT*

c_id – *DOESNT_EXISTS* – אם לא קיים לקוח בעל c_id .

c_id – *SUCCESS* – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(1)$ בממוצע על הקלט.

StatusType buyRecord(int c_id, int r_id)

הפונקציה מוסיפה רכישה אחת לתקליט ובמידה והלקוח הינו חבר מועדון מוסיפה להוצאות החודשיות שלו את מחיר התקליט שנקבע על ידי הנוסחה הבאה: $100 + t$. כאשר t הינו מספר הרכישות שבוצעו לתקליט זה לפני קניה זו.

פרמטרים: c_id – המזהה של הלקוח, r_id – המזהה של התקליט.

ערך החזרה:

$c_id < 0$ או $r_id < 0$ – *INVALID_INPUT*

c_id – *DOESNT_EXISTS* – אם לא קיים לקוח בעל c_id או $r_id \geq \text{number_of_records}$

c_id – *SUCCESS* – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(\log n)$.

StatusType addPrize(int c_id1, int c_id2, double amount)

הפונקציה מוסיפה פרס לחברי המועדון בעלי c_id המקיים $c_id1 \leq c_id < c_id2$ (ייתכן כי c_id1, c_id2 אינם מזהים הנמצאים במערכת). הפרס הינו בגודל $amount$ והוא יורד מההוצאות החודשיות של כל מי שזכה בו.

פרמטרים: c_id1, c_id2 – טווח מזהי הזוכים, $amount$ – גודל הפרס.

ערך החזרה:

$c_id1 < 0$, $c_id1 < c_id2$ או $c_id2 \leq 0$ – *INVALID_INPUT*

c_id – *SUCCESS* – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(\log n)$.

Output_t<double> geptExpenses(int c_id)

הפונקציה מחזירה את סך ההוצאות החודשיות של חבר המועדון בעל c_id (ההוצאות כוללות גם את ההוצאות בעקבות הפרסים).

פרמטרים: c_id – המזהה של הלקוח.

ערך החזרה:



$c_id < 0$ – *INVALID_INPUT*

c_id – אם לא קיים חבר מועדון בעל c_id . *DOESNT_EXISTS*

במידה והפעולה הצליחה, מוחזר סך ההוצאות של הלקוח.

סיבוכיות זמן: $O(\log n)$.

StatusType **putOnTop**(int r_id1 , int r_id2)

הפונקציה עורמת את הערמה שבה נמצא התקליט r_id1 על הערמה שבה נמצא r_id2 . כאשר הגובה של כל תקליט בערמה של r_id1 גדל בגובה של הערמה של r_id2 והעמודה שבה נמצאים היא באינדקס של האיבר התחתון ביותר בערמה של r_id2 .

פרמטרים: r_id1, r_id2 – המזהים של תקליטים.

ערך החזרה:

$r_id1, r_id2 < 0$ – *INVALID_INPUT*

$r_id1, r_id2 \geq \text{number_of_records}$ – אם *DOESNT_EXISTS*

r_id1 או r_id2 כבר אחד מעל השני. *FAILURE*

SUCCESS – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(\log^* m)$.

StatusType **getPlace**(int r_id , int $*column$, int $*hight$)

הפונקציה מחזירה את העמודה והגובה של התקליט r_id במשתנים $column, hight$ בהתאמה.

פרמטרים: r_id – המזה של תקליט, $column, hight$ – משתנים שיכילו את ערכי החזרה.

ערך החזרה:

$r_id < 0$ או $column, hight = NULL$ – *INVALID_INPUT*

$r_id \geq \text{number_of_records}$ – אם *DOESNT_EXISTS*

SUCCESS – במידה והפעולה הצליחה.

סיבוכיות זמן: $O(\log^* m)$.

הנחיות:

חלק יבש:

- החלק היבש הוא חלק מהציון על התרגיל כפי שמצוין בנהלי הקורס.
- 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 תגרור 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. חלק יבש זה לא תיעוד קוד.
- 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 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!

מבני נתונים 234218 חורף תשפ"ג

גיליון רטוב מספר 2 – מעודכן לתאריך 29.05.2023
עמוד 5 מתוך 6



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**, ב-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 הקוד להתקמפל על csl3 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.



הערות נוספות:

- חתימות הפונקציות שעליכם לממש ומספר הגדרות נמצאים בקובץ `recordsCompany.h`.
- קראו היטב את הקובץ הנ"ל, לפני תחילת העבודה.
- אין לשנות את הקבצים `mainWet2.cpp` ו-`utilesWet2.h` אשר סופקו כחלק מהתרגיל, ואין להגיש אותם.
 - את שאר הקבצים ניתן לשנות.
 - תוכלו להוסיף קבצים נוספים כרצונכם, ולהגיש אותם.
 - העיקר הוא שהקוד שאתם מגישים יתקמפל עם הפקודה לעיל, כאשר מוסיפים לו את שני הקבצים `mainWet2.cpp` ו-`utilesWet2.h`.
- 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.
- בפרט, אסור להשתמש ב-`std::pair`, `std::vector`, `std::iterator`, או כל אלגוריתם של STL.
- ניתן להשתמש במצביעים חכמים (Smart pointers כמו `shared_ptr`), בספריית `math` בספריית `exception`.
- חשוב לוודא שאתם מקצים/משחררים זיכרון בצורה נכונה (מומלץ לוודא עם `valgrind`). לא חייבים לעבוד עם מצביעים חכמים, אך אם אתם מחליטים כן לעשות זאת, לוודא שאתם משתמשים בהם נכון. (תזכרו שהם לא פתרון קסם, למשל, כאשר יוצרים מעגל בהצבעות)
- שגיאות של `ALLOCATION_ERROR` בד"כ מעידות על זליגה בזיכרון.
- מצורפים לתרגיל קבצי קלט ופלט לדוגמא, ניתן להריץ את התוכנה על הקלט ולהשוות עם הפלט המצורף.
- **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:

- בתיקייה הראשית:
 - קבצי ה-Source Files שלכם. למעט הקבצים `mainWet2.cpp` ו-`utilesWet2.h`, שאסור לשנות.

מבני נתונים 234218 חורף תשפ"ג

גיליון רטוב מספר 2 – מעודכן לתאריך 29.05.2023
עמוד 6 מתוך 6



- קובץ PDF בשם dry.pdf אשר מכיל את הפתרון היבש. 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:

John Doe 012345678 doe@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 ומערכת הבדיקה האוטומטית 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!
- אחרי שאתם מכינים את ההגשה בקובץ zip מומלץ מאוד לקחת אותה לשרת ולהריץ את הבדיקות שלכם עליה כדי לוודא שאתם מגישים את הקוד שהתכוונתם להגיש בדיוק (ושהוא מתקמפל).

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 למתרגל האחראי בלבד בכתובת turotovsally@campus.technion.ac.il. 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!