

קורס: מבני נתונים 1 - 234218

חלק יבש לרטוב 2

מגישות:

1. אמילי דרור [324934082]
2. עדי צח [207844358]

תאריך הגשה: 17.06.2023

**תיאור מבנה הניתונים**

מחלקת RecordsCompany

טבלת ערבול [customers\_id] – טבלת ערבול מסוג Chain Hashing אלא שבמקום רשימות מקושרות תחזיק עצי AVL אשר יכילו מצביעים ללקוחות ע״פ המזהי ID שלהם.

עץ דרגות [prises] – עץ avl המחזיק צמתים המתייחסים לטווחי ID של לקוחות. מדובר בעץ דרגות כיוון שבכל צומת בעץ יתוחזק פרמטר פרס.

UnionFind [records] – המבנה יחזיק את כלל התקליטים במערכת. נממש את המבנה בעזרת עצים הפוכים עם איחוד לפי גודל וכיווץ מסלולים. נדאג לתחזק את פרמטר הגובה של כל תקליט דרך המבנה.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Costumer

שדות:

* id - מזהה ייחודי
* clubMember - שדה בוליאני שמציין אם המשתמש הינה חבר מועדון.
* phone – מספר הטלפון של המשתמש
* monthlyPatments – חובות חודשיים למשמתמש במידה והנ״ל הינו חבר מועדון.
* כסף??????

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Records

שדות:

* id - מזהה ייחודי
* copies - שדה שמציין את מספר העותקים של התקליט
* buys – כמות הפעמים שהתקליט נקנה

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**­­­­­­**

**תיאור המתודות של RecordsCompany:**

RecordsCompany()

*תיאור:* . נאתחל את כלל המבנים הקיימים במערכת (עץ דרגות, UnionFind, HashTable) שלנו עם ערכי null.

*סיבוכיות: .* O(1)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

~RecordsCompany()

*תיאור:* נמחוק את המידע המוחזק בעץ, ב-UnionFind וב-HashTable.

*סיבוכיות:* O(n+m)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType newMonth(int \*records\_stocks, int number\_of\_records)

*תיאור: נאפס את סידור התקליטים ואת סכום הכסף שכל אדם חייב. נייצר פה את המבנה unionfind לתקליטים כמו שצריך!!*

*סיבוכיות: ייצור אובייקט הכנסה לשלוש עצי avl (נשים לב שגדלי בעצים חסומים ע״י מספר הסרטים הכולל במערכת).*

*סיבוכיות:* O(n+m)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType addCostumer(int c\_id, int phone)

*תיאור: ננסה להוסיף את הלקוח הנוסף למערכת ע״י צירופו ל-HashTable* O(1) משוערך בממוצע על הקלט*, נבחן מקרי בקצה ובמקרה והקלט תקין נייצר אובייקט חדש מסוג costumer* O(1)*.*

*סיבוכיות: O*(1) משוערך בממוצע על הקלט

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Output\_t<int> getPhone(int c\_id)

*תיאור: נגש לללקוח בעל המזהה המתקבל בעזרת טבלת הערבול שלנו O*(1) בממוצע ונחזיר את מספר הטלפון של האובייקט.

*סיבוכיות: O*(1) בממוצע על הקלט

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType makeMember(int c\_id)

*תיאור: נגש ללקוח בעל המזהה המתקבל בעזרת טבלת הערבול שלנו בסיבוכיות O(log(n)) זאת כיוון שטבלת הערבול שלנו ממומשת עם עצי avl ולא רשימות מקושרות (במקרה הגרוע ביותר כלל n הלקוחות ימוקמו באותו עץ ה-avl ועל כל הסבוכיות הינה O(log(n))) נהפוך את הלקוח לחבר מועדון ע״י שינוי השדה הרלוונטי אצלו O(1).*

*סיבוכיות: O(O(log(n))*

*סיבוכיות: הוצאה מעצי avl .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Output\_t<bool> isMember(int c\_id)

*תיאור: נגש ללקוח בעל המזהה המתקבל בעזרת טבלת הערבול שלנו בסיבוכיות O*(1) בממוצע על הקלט*, נחזיר את שדה חבר-מועדון של הלקוח O(1).*

*סיבוכיות: O*(1) בממוצע על הקלט

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType buyRecord(int c\_id, int r\_id)

*תיאור: נעזר במבנה ה-UnionFind בשביל למצוא את התקליט הרלוונטי סיבוכיות O(log\*(m)). נעזר בhashtable על מנת לגשת ללקוח הרלוונטי בסיבוכיות O(log(n)) זאת כיוון שטבלת הערבול שלנו ממומשת עם עצי avl ולא רשימות מקושרות (במקרה הגרוע ביותר כלל n הלקוחות ימוקמו באותו עץ ה-avl ועל כל הסבוכיות הינה O(log(n))). כעת במידה והלקוח הינו חבר מועדון אזי נבחן את כמות הקניות שבוצעו לתקליט לפני פועלה זו ונוסיף לתשלום החודשי של הלקוח את סכום הקניות הנ״ל + 100, לאחר מכן נוסיף 1 לכמות הקניות שבוצעו לתקליט זה בסיבוכיותO(1) . אם הלקוח אינו חבר מועדון אזי רק נוסיף 1 לכמות הקניות שבוצעו לתקליט בסיבוכיותO(1) .*

*סיבוכיות: סה״כ O(log(n))).*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType addPrize(int c\_id1, int c\_id2, double amount)

*תיאור: ננסה להכניס לעץ הדרגות (אשר מתייחס לפרסים אשר כל id זכאי להם) שני צמתים המתייחסים לגבולות ה-id שהתקלבו כפרמטרים (במידה וקיים כבר צומת עם 1 מערכי ה-id שהתקבלו אזי לא נוסיף את הצומת הרלוונטי לו, כן ננסה לבצע את תהליך ההכנסה). תהליך ההכנסה הינו כדלקמן, בעת פניהשינחהשלנהלשנש בעת פניהה שמאלה, בן ימני לךצמשכךלשמצ*

*סיבוכיות: סה״כ O(log(n))).*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Output\_t<double> getExpenses(int c\_id)

*תיאור: נעזר בhashtable על מנת לגשת ללקוח הרלוונטי בסיבוכיות O(log(n)) זאת כיוון שטבלת הערבול שלנו ממומשת עם עצי avl ולא רשימות מקושרות (במקרה הגרוע ביותר כלל n הלקוחות ימוקמו באותו עץ ה-avl ועל כל הסבוכיות הינה O(log(n))). לאחר מציאת הלקוח נבחן את שדה התשלום-החודשי שלו. נעזר בעץ הפרסים על מנת לאפיין בסיבוכיות של O(log(n))) את כמות הנוזלות להם זכאי הלקוח כתוצאה מכלל הפרסים הרלוונטיים לו. נסכום את שדה התשלום החודשי של הלקוח וסכום הפרסים להם הוא זכאי ונחזיר מספר זה (יכול להיוצת מספר שלילי<?).*

*סיבוכיות: סה״כ O(log(n))).*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType putOnTop(int r\_id1, int r\_id2)

*תיאור: נבצע איחוד קבוצות ועדכון פרמטר נלווה עבור מבנה ה-UnionFind בהתאם לנלמד בתרגול. להוסיף*

*סיבוכיות:*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType getPlace(int r\_id, int \*column, int \*hight)

*תיאור: נעזר במבנה ה-UnionFind על מנת לאפיין את הגובה והשורה. להוסיף*

*סיבוכיות:*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*