**פרויקט 18 – ניהול ספריית האוניברסיטה**

**מטרת הפרויקט:**

מטרת הפרויקט הייתה לנהל את ספריית האוניברסיטה בעזרת מבנה נתונים יעיל ככל הניתן התומך בפעולות השונות שהוגדרו בממ"ן.

**תיאור מבנה הנתונים:**

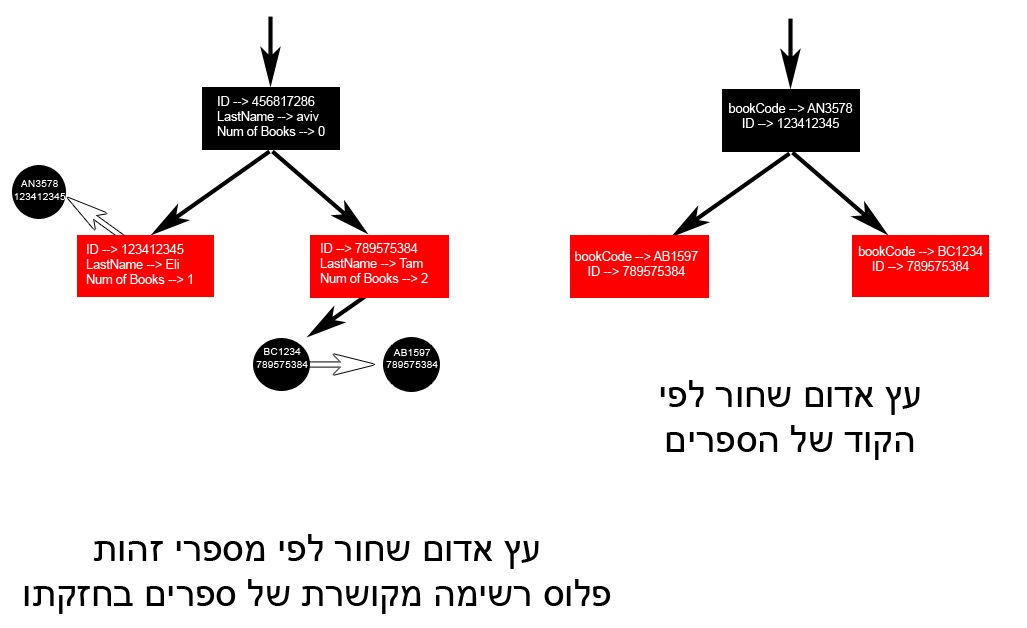
נשתמש בעץ אדום שחור אחד בעל המפתח של מספרי הזהות.

עץ זה יכיל בתוכו את השדות מספר זהות,שם משפחה,כמות הספרים שברשותו, ורשימה מקושרת אשר מכילה את כל הספרים שברשותו ( הרשימה תכיל עד גג 10 ספרים ).

נשתמש בעץ אדום שחור שני בעל המפתח של קוד הספרים.

עץ זה יכיל בתוכו את השדות קוד הספר, ומספר הזהות.

להלן תיאור גרפי של מבנה הנתונים:



**בדיקות הרצה:**

בוצעו שני הרצות על 25 קלטים שונים, ועליהם ביצאנו את הבדיקות הבאות:

* הדפסת העץ אדום שחור בעל מפתח תעודות הזהות בסדר ממוין.
* הדפסת העץ אדום שחור בעל מפתח קוד הספרים בסדר ממוין.
* בדיקה אצלי מי נמצא הספר שהקוד שלו הוא X (קוד כלשהו).
* בדיקה של כמה ספרים יש אצל האדם שמספר תעודת הזהות שלו היא Y (מספר כלשהו).
* בדיקה מי הם האנשים בעלי מספר הרב ביותר של ספרים בחזקתם.
* בדיקה של איזה ספרים יש אצל אדום שמספר תעודת הזהות שלו היא Y (מספר כלשהו).
* הסרת ספר בעל קוד מסויים.
* הסרת משתמש מהמבנה (כמובן הסרת כל הספרים בהתאם).

**סיבוכיות זמן הריצה של האלגוריתם:**

נגדיר:

מספר הספרים - m, מספר האנשים - n.

T - עץ אשר המפתח שלו תעודות זהות, S - עץ אשר המפתח שלו קוד הספרים.

**הערות**

* השגרות של העץ אדום שחור נכתבו ע"פ הספר ולכן זמן ריצתם כזמן הריצה המוכח בספר.
* הוספת ספר למשתמש מעלה את המשתנה numOfBooks ב 1, והסרת ספר מהמשתמש מורידה את numOfBooks ב 1, שינויים אלו מתבצעים בזמן קבוע.

הכנסת ספר - insertBook

1. חיפוש האם האדם כבר קיים בעץ T ( lg n ).
2. אם כן --> הוספת הספר לרשימה המקושרת (O(1)) והוספת הספר לעץ S (lg m)
3. אם לא --> הוספת איבר לעץ T והוספת איבר לעץ S ( O(lg m + lg n) ).

סה"כ זמן ריצה : - הכנסות לעץ אדום שחור.

מחיקת ספר - removeBook

1. חיפוש האם קיים הספר בעץ S ( lg m ).
2. אם לא --> סיימנו.
3. אם כן --> מחיקת הספר מעץ S ( lg m) , חיפוש התעודת זהות המתאימה בעץ T (lg n) והורדת הספר מהרשימה המקושרת שם ( O(1) ).

סה"כ זמן ריצה : - חיפוש ומחיקה מעץ אדום שחור .

הכנסת משתמש לספריה - insertMember

1. הכנסת המשתמש לעץ T ( lg n ).

סה"כ זמן ריצה : - הכנסה לעץ אדום שחור.

הסרת משתמש מהספריה - removeMember

1. חיפוש האם האדם כבר קיים בעץ T ( lg n ).
2. אם לא --> סיימנו.
3. אם כן --> מעבר על רשימת כל הספרים שיש לאותו אדם (יש מקסימום 10 ספרים לאדם, ולכן הזמן קבוע O(1) ), ומחיקת כל ספר מעץ S ( lg m ).
4. לבסוף - מחיקת המשתמש עצמו מעץ T ( lg n ).

סה"כ זמן ריצה : - חיפוש ומחיקה מעץ אדום שחור .

המשתמשים עם הכי הרבה ספרים בחזקתם - mostBook

1. מציאת העלה הקטן ביותר בעץ T ( lg n ).
2. מעבר על העץ על מנת למצוא מהו המספר הרב ביותר של ספרים שקיימים אצל אדם כלשהו ( O(n) )
3. מעבר נוסף על העץ על מנת למצוא מיהם כל האנשים אשר בעלי המספר הרב ביותר של ספרים שמצאנו מקודם ( O(n) ).

סה"כ זמן ריצה : - מעבר בעץ אדום שחור .

בדיקת כמה ספרים בחזקתו של אדם howManyBooks

1. חיפוש האם האדם כבר קיים בעץ T ( lg n ).
2. אם לא --> סיימנו.
3. אם כן --> החזרת מספר הספרים בחזקתו (אנו שומרים ערך זה במשתנה! ).

הדפסת הספרים אשר נמצאים בחזקתו של אדם כלשהו whichBooks

1. חיפוש האם האדם כבר קיים בעץ T ( lg n ).
2. אם לא --> סיימנו.
3. אם כן --> הדפסת רשימה (זמן קבוע O(1) כי יש במקסימום 10 ספרים לאדם! )

סה"כ זמן ריצה : - חיפוש בעץ אדום שחור.

מציאת האדם אשר מחזיק את הספר whoHasThisBook

1. חיפוש האם קיים הספר בעץ S ( lg m ).

סה"כ זמן ריצה : - חיפוש בעץ אדום שחור.