<u>מבני נתונים 2023- תרגיל 7</u>

<u>חלק א'</u>

שאלה 1

בשאלה זו נרצה לעקוב אחרי מבנהו של עץ AVL המכיל מספרים שלמים. ומצב הקודקודים בו במהלך הכנסה והוצאה של איברים. איירו או תארו בדרך ברורה את מצב העץ לאחר כל פעולה (אתם יכולים לחקות את הדוגמאות בתרגול). במידה ונדרשת פעולת רוטציה לשמירת האיזון, הראו קודם את המצב לאחר פעולת ההכנסה/מחיקה ולפני הרוטציה, ציינו איזה קודקוד יצא מאיזון ואיזו רוטציה צריך להפעיל, ואז הראו את המצב לאחר הפעלת הרוטציה. אין צורך להראות את רוטציות LR,RL בשני מהלכים, פשוט הראו מה המצב אחריהן.

בכלל הסעיפים, נתחיל עם עץ ריק ונבצע את הפעולות לפי הסדר משמאל לימין.

1. insert(1),insert(2),insert(3),insert(4),insert(5),insert(6),insert(7)
2. insert(1),insert(7),insert(2),insert(6),insert(3),insert(5),insert(4)
3. insert(4),insert(2),insert(5),insert(1),insert(3),delete(4)
4. insert(1), insert(3), insert(5), insert(2), delete(5), insert(−1), delete(2), insert(2), delete(−1), insert(4), delete(1)

<u>חלק ב'</u>

שאלה 2

הציגו מבנה נתונים אשר יכיל איברים מהתחום $\{0, \dots, k-1\}$ עבור $k \epsilon \mathrm{N}$, ללא חזרות. n יהיה מספר האיברים ברגע מסוים.

מבנה הנתונים יממש את הממשק הבא:

- $insert(m)$: הכנסת איבר בעל המפתח m בזמן **ממוצע** של $O(logn)$.
- $delete(m)$: מחיקת האיבר בעל המפתח m בזמן **ממוצע** של $O(logn)$. במידה והמפתח m לא קיים במבנה, הפונקציה תחזיר שגיאה.
- $find(m)$: בדיקה האם המפתח m קיים במבנה הנתונים, בזמן **ממוצע** של $O(1)$.
- $getMin(), getMax()$ - מציאת ערכי מינימום ומקסימום **במקרה הגרוע ביותר** ב$O(1)$.
- $successor(m)$ - מציאת האיבר הבא בגודלו לm בזמן **ממוצע** של $O(1)$. במידה m לא קיים במבנה או שהוא המקסימלי הפונקציה תחזיר שגיאה.
- $predecessor(m)$ - מציאת האיבר הקודם בגודלו לm בזמן **ממוצע** של $O(1)$. במידה m לא קיים במבנה או שהוא המינימלי הפונקציה תחזיר שגיאה.
- $byOrder()$- החזרת מערך ממוין בסדר עולה המכיל את האיברים במבנה. **במקרה הגרוע ביותר** ב $O(n)$.

חלק ג'

שאלה 3:

כתבו אלגוריתם אשר מקבל שורש של עץ חיפוש בינארי וקובע האם העץ מקיים את תכונת הAVL (גורם האיזון קטן או שווה ל1 עבור כל קודקוד בעץ). נתחו את זמן הריצה (במקרה הגרוע).

שאלה 4

תארו בקצרה מבנה נתונים הממשמש את הממשק הבא:

- המבנה יכיל n איברים שכל אחד מהם מורכב ממפתח $key\epsilon\{0,1,..,k-1\}$ עבור $k\epsilon\mathbb{N}$ כלשהו ללא חזרות. וערך $value\epsilon\{0,1,..,v-1\}$ עבור $v\epsilon\mathbb{N}$. כלומר, כל איבר הוא הצמד ($key, value$).
- $insert((key, value))$: הכנסת איבר בעל המפתח key וערך $value$ בזמן **ממוצע** של $O(1)$, ובמקרה **הגרוע** ב$O(logn)$.
- $delete((key, value))$: מחיקת האיבר בעל המפתח key וערך $value$ בזמן **ממוצע** של $O(1)$, ובמקרה **הגרוע** ב$O(logn)$. במידה והמפתח key לא קיים במבנה, הפונקציה תחזיר שגיאה.
- $find((key, value))$: בדיקה האם המפתח key קיים במבנה הנתונים, אם כן, החזרת הערך $value$ המתאים בזמן **ממוצע** של $O(1)$ ובמקרה הגרוע ב$O(logn)$.

אין צורך בהוכחת נכונות פורמלית, בניגוד לשאלות אחרות, אין צורך לתאר מימוש כל פונקציה באופן ספציפי אלא לתת רק סקירה כללית של מבנה הנתונים ולהסביר מדוע המבנה אכן עומד בדרישות.

שאלה 5

שרטטו avl המקיים את התכונות הבאות:

- לעץ יש 11 קודקודים.
- אם נמחק מהעץ כל זוג קודקודים אפשרי (אחד אחרי השני) גובה העץ יישאר זהה.
- אם נוסיף קודקוד (כל קודקוד אפשרי), גובה העץ יישאר זהה.