

## מבני נתונים – פרויקט חקר תכנותי מספר 1 – עץ דרגות

### הקדמה:

בתרגיל זה שני חלקים:

- חלק המעשי: מימוש של List באמצעות עץ AVL. עמודים 1-2 במסמך זה מתארים את החלק הזה.
  - חלק ניסויי-תיאורטי: בהתבסס על המימוש מהחלק המעשי, נבצע מספר "ניסויים" עם ניתוח תיאורטי נלווה ושאלות חקר. עמודים 3-5 מתארים את החלק הזה.
- שימו לב:** בסוף המסמך (עמוד 5) ישנן הוראות הגשה – הקפידו לפעול לפיהן. **תאריך הגשה: 24.4.** בנוסף, יש לעקוב אחר השרשרת הנועץ בפורום בו נפרסם הבהרות חשובות.

### חלק מעשי

#### דרישות

בתרגיל זה נממש את ה ADT **רשימה** באמצעות עץ AVL. לכל איבר ברשימה יש ערך (info). המימוש יהיה בשפת **python 3.9** וצריך להיות מבוסס על קובץ השלד המופיע באתר הקורס. הפעולות שיש לממש הן:

פעולה	תיאור
empty()	הפונקציה מחזירה ערך TRUE אם ורק אם הרשימה ריקה.
retrieve(i)	הפונקציה מחזירה את ערך האיבר במקום ה-i אם קיים, אחרת היא מחזירה None.
insert(i, s)	הכנסת איבר בעל ערך s לרשימה במקום ה-i, במידה וקיימים לפחות i איברים ברשימה. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ בשלב תיקון העץ על מנת לשמר את תכונת האיזון.
delete(i)	מחיקת האיבר במקום ה-i ברשימה, אם הוא קיים. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסך הכל בשלב תיקון העץ על מנת לשמר את תכונת האיזון. אם לא קיימים מספיק איברים ברשימה הפונקציה מחזירה 1 –.
first()	מחזירה את ערך האיבר הראשון ברשימה, או None ברשימה ריקה.
last()	מחזירה את ערך האיבר האחרון ברשימה, או None ברשימה ריקה.
listToArray()	הפונקציה מחזירה מערך המכיל את איברי הרשימה לפי סדר האינדקסים, או מערך ריק אם הרשימה ריקה.
length()	הפונקציה מחזירה את מספר האיברים ברשימה.
split(i)	הפונקציה מקבלת אינדקס שנמצא ברשימה. על הפונקציה לחצות אותה לשתי רשימות, לפני האינדקס i ואחרי האינדקס i (פירוט בקובץ שלד). יש לממש את הפונקציה על פי המימוש שנלמד בהרצאה לפיצול בעץ AVL בסיבוכיות $O(\log n)$ .
concat(lst)	הפונקציה מקבלת רשימה. על הפונקציה לשרשר אותה אל סוף הרשימה הנוכחית. על הפעולה לרוץ בזמן $O(\log n)$ . יש להחזיר את הערך המוחלט של הפרש הגבהים של עצי ה AVL שמוזגו.
search(val)	החזרת האינדקס הראשון ברשימה בו מופיע הערך val, או 1 – אם לא קיים כזה.

בנוסף למימוש הפונקציות האלו, יש לממש את מחלקת **AVLNode** כפי שמתואר בקובץ. מטעמי נוחות, נדרוש שלכל עלה יהיו 2 בנים "וירטואליים", כלומר, צמתים שלא מייצגים איברים במבנה הנתונים. באופן זה, נוח יותר לממש גלגולים מכיוון שלכל צומת יהיו 2 בנים וזה חוסך טיפול במקרי קצה.

למחלקה AVLNode יש את המתודות הבאות (המפרט המלא נמצא בקובץ השלד):

getHeight – מחזיר את הגובה של הצומת, או 1 – אם הצומת הוא וירטואלי.

getValue – מחזיר את info של הצומת או None אם הצומת הוא וירטואלי.

getLeft – מחזיר את הבן השמאלי של הצומת, או None אם אין כזה.

getRight – מחזיר את הבן הימני של הצומת, או None אם אין כזה.

getParent – מחזיר את ההורה של הצומת, או None אם אין כזה.

isRealNode – מחזיר TRUE אם הצומת מייצג צומת אמיתי בעץ (קרי: צומת שאינו וירטואלי).

## הערות חשובות:

1. **המימוש יבוצע על ידי מילוי קובץ השלד. במידת הצורך, ניתן להרחיב את המימוש** (למשל להוסיף פונקציות עזר שאינן מופיעות בשלד), אך **אסור לשנות את הגדרות הפונקציות לעיל**. על כל הפונקציות/מחלקות להופיע בקובץ יחיד.
2. **אין להשתמש באף מימוש ספרייה של מבנה נתונים.**

## **סיבוכיות**

יש לתעד בקוד ובמסמך נפרד (ביותר פירוט) את סיבוכיות זמן הריצה במקרה הגרוע (האסימפטוטית, במונחי O הדוקים) של כל פונקציה **שמכילה לולאות/רקורסיה**, כתלות במספר האיברים בעץ n. עליכם להשיג סיבוכיות זמן ריצה (במקרה הגרוע ביותר) נמוכה ככל הניתן עבור כל אחת מהפונקציות.

## **פלט**

אין צורך בפלט למשתמש.

## **תיעוד**

בנוסף לבדיקות אוטומטיות של הקוד שיוגש, קובץ המקור ייבדק גם באופן ידני. חשוב להקפיד על תיעוד לכל פונקציה, וכמות סבירה של הערות. **הקוד צריך להיות קריא**, בפרט הקפידו על בחירת שמות משתנים ועל אורך השורות.

יש להגיש בנוסף לקוד גם מסמך תיעוד חיצוני. המסמך יכול את תיאור המחלקה שמומשה, ואת תפקידו של כל חבר במחלקה. עבור כל פונקציה במחלקה יש לפרט מה היא עושה, כיצד היא פועלת ומה **סיבוכיות זמן הריצה שלה**. בפרט, אם פונקציה קוראת לפונקציית עזר, יש להתייחס גם לפונקציית העזר בניתוח. עבור פונקציות שעולות זמן קבוע יספיק להביא רק תיאור קצר ולא לפרט את ניתוח הסיבוכיות.

## **בדיקות**

התרגילים ייבדקו באמצעות תוכנת טסטר שקוראת לפונקציות המפורטות מעלה בתרחישים שונים, ומוודאת את נכונות התוצאות. קובץ הטסטר שלנו **לא יפורסם** לפני הבדיקות. **מומלץ מאוד לממש אוסף בדיקות עבור המימוש**, לא בשביל ההגשה, אלא כדי לבדוק שהקוד לא רק רץ, אלא גם נכון!

בקובץ שתגישו **לא תהיה פונקציית main ולא יהיו הרצות קוד/הדפסות**, דבר זה יפגע בטסטר שיבדוק לכם את התרגילים.

## חלק ניסויי/תיאורטי

### שאלה 1:

בשאלה זאת נדון בכמות צעדי האיזון הממוצעת הנדרשת בסדרת הכנסות, בסדרת מחיקות ובסדרה מעורבת של הכנסות ומחיקות.

- לצורך הניתוח, נבנה עצי AVL בגדלים שונים. מספר איברים שנכניס לעץ יהיה  $n = 1000 \cdot 2^i$  כאשר  $i=1, \dots, 10$ . כלומר, עבור  $i=1$  העץ בגודל 2000, ועבור  $i=10$  העץ בגודל כמיליון.
- לכל גודל של עץ, נבצע 3 ניסויים נפרדים:
  - נכניס איברים בסדר אקראי.
  - נכניס איברים בסדר אקראי (לא נספר), ולאחר מכן נמחק אותם בסדר אקראי.
  - נכניס  $n/2$  איברים בסדר אקראי, לאחר מכן נבצע  $n/4$  הכנסות ומחיקות אקראיות לסירוגין.

1. עבור כל ניסוי, יש לציין את מספר פעולות האיזון שנדרשו כדי לתקן את העץ עבור הפעולות שהודגשו.

מספר סידורי $i$	ניסוי 1 - הכנסות	ניסוי 2 - מחיקות	ניסוי 3 - הכנסות ומחיקות לסירוגין
1			
2			
3			
...			
10			

2. היעזרו באקסל, איזה ביטוי אסימפטוטי תואם כל עמודה? (במונחים של  $n$ )  
**הדרכה:** במקרה שמצפים לביטוי  $f(n)$ , ניתן לבדוק את מידת ההתאמה האמפירית של הנתונים על-ידי חילוק הנתונים ב  $f(n)$ . בתוכנת אקסל, למשל, ניתן לחשב קו-מגמה (trendline) ומדד ה- $R^2$  מעיד על איכות הקירוב.

### שאלה 2:

בשאלה זו נרצה לנתח את העלות של פעולות ה-join המתרחשות במהלך ביצוע split. נגדיר את העלות של פעולת join בודדה כערך המוחלט של הפרש הגבהים של העצים שאוחדו.

- לצורך הניתוח, נבנה עצי AVL בגדלים שונים. מספר האיברים שנכניס לעץ יהיה  $n = 1000 \cdot 2^i$  כאשר  $i=1, \dots, 10$ . כלומר, עבור  $i=1$  העץ בגודל 2000, ועבור  $i=10$  העץ בגודל כמיליון. את האיברים יש להכניס **בסדר אקראי**.
- לכל גודל של עץ, נבצע 2 ניסויים נפרדים (ולכן נבנה שני עצים עם אותו סדר הכנסה, לכל גודל):
  - בניסוי אחד נבצע split על אינדקס אקראי ברשימה.
  - בניסוי שני נבצע split על האינדקס המקסימלי בתת העץ השמאלי של השורש.

1. תעדו בטבלה שלהלן את העלות הממוצעת של פעולות ה-join ואת העלות של פעולת ה-join היקרה ביותר.

מספר סידורי $i$	עלות join ממוצע עבור split אקראי	עלות join מקסימלי עבור split אקראי	עלות join ממוצע עבור split של האיבר מקסימלי בתת העץ השמאלי	עלות join מקסימלי עבור split של איבר מקסימלי בתת העץ השמאלי
1				
...				

				10
--	--	--	--	----

2. נתחו באופן תיאורטי את העלות של **join ממוצע לשני התרחישים** (split אקראי או על האיבר המסוים שבחרנו), והסבירו אם התוצאות מתיישבות עם ניתוח הסיבוכיות התאורטי.

3. נתחו באופן תיאורטי את העלות של **join מקסימלי בתרחיש אחד** של split על האיבר המסוים שבחרנו, והסבירו אם התוצאות מתיישבות עם ניתוח הסיבוכיות התאורטי.

### שאלה 3:

בניסוי זה נשווה את ביצועי עץ AVL לעומת עץ חיפוש בינארי רגיל (לא מאוזן). ממשו רשימה באמצעות עץ לא מאוזן שכזה, למשל ע"י הסרת הגלגולים מהמימוש הקיים המשתמש בעץ AVL (אין צורך להגיש את המימוש הזה, אבל תהליך הבדיקה עשוי לכלול הסרת הגלגולים בקוד שלכם ע"י הבדקים כדי לוודא שהתוצאות שציינתם תואמות את הקוד שלכם).

עבור כל  $i = 1, \dots, 10$  חזרו על התהליך הבא:

- הכניסו  $n = 1000 * i$  איברים לרשימה הממומשת בעזרת עץ AVL (לפי שלושת המקרים המפורטים בהמשך).
- הכניסו את אותם האיברים לרשימה הממומשת בעזרת עץ חיפוש לא מאוזן.
- עבור כל אחד מהעצים, רשמו את מספר פעולות האיזון הממוצע ועומק הצומת המוכנס הממוצע פר הכנסה (הממוצע על פני  $n$  ההכנסות).

חזרו על התהליך כאשר  $i = 1000$  ההכנסות הן לפי:

1. **הכנסות להתחלה** – סדרת הכנסות לראש הרשימה.
2. **הכנסות מאוזנות** – סדרת הכנסות המניבה עץ בעל הגובה המינימלי האפשרי גם ללא מנגנון איזון – כלומר שגובה העץ הוא  $\lfloor \log n \rfloor$  (כאשר  $n = 1000 * i$ ). עליכם לחשוב על סדרת אינדקסים שגורמת לכך.
3. **סדרה אקראית** - מכניסים במקומות אקראיים מתוך גודל הרשימה הנוכחי.

רשמו את התוצאות בטבלאות הבאות:

מספר פעולות האיזון בממוצע	עץ AVL חשבונית	עץ ללא מנגנון איזון חשבונית	עץ AVL סדרה מאוזנת	עץ ללא מנגנון איזון סדרה מאוזנת	עץ AVL סדרה אקראית	עץ ללא מנגנון איזון סדרה אקראית
1						
2						
...						

מספר פעולות האיזון בממוצע	עץ AVL חשבונית	עץ ללא מנגנון איזון חשבונית	עץ AVL סדרה מאוזנת	עץ ללא מנגנון איזון סדרה מאוזנת	עץ AVL סדרה אקראית	עץ ללא מנגנון איזון סדרה אקראית
1						
2						
...						

מה הייתם מצפים שתהיינה התוצאות, והאם התוצאות האמיתיות מסתדרות עם ציפייה זו? הסבירו.

### הוראות הגשה

הגשת התרגיל תתבצע באופן אלקטרוני באתר הקורס במודל.

#### הגשת התרגיל היא בזוגות בלבד!

כל זוג יבחר נציג/ה ויעלה רק תחת שם המשתמש של הנציג/ה את קבצי התרגיל (תחת קובץ zip) למודל.

על ההגשה לכלול שלושה קבצים:

1. קובץ המקור (הרחבה של קובץ השלד שניתן) תחת השם AVLTreeList.py.
2. קובץ טקסט info.txt המכיל את פרטי הזוג: מספר ת"ז, שמות, ושמות משתמש.
3. מסמך תיעוד חיצוני, המכיל גם את תוצאות המדידות. את המסמך יש להגיש באחד הפורמטים הבאים: doc, docx או pdf.

שמות קובץ התיעוד וקובץ הק zip צריכים לכלול את שמות המשתמש האוניברסיטאיים של **הזוג המגיש** לפי הפורמט AVLTreeList\_username1\_username2.pdf/doc/zip .... בתוכן הקבצים יש לציין את שמות המשתמש, תעודות הזהות ושמות המגשים (בכותרת המסמך ובשורת הערה בקובץ המקור).

הגשת שיעורי הבית באיחור - באישור מראש בלבד. הגשה באיחור ללא אישור תגרור הורדת נקודות מהציון.  
**הגשת התרגיל היא חובה לשם קבלת ציון בקורס.**

**בהצלחה!**