אוניברסיטת תל אביב

סמסטר ב' תשפ"ג

מבני נתונים – פרויקט מספר 1 – עץ דרגות

הקדמה:

בתרגיל זה שני חלקים:

1. חלק המעשי: מימוש של עץ AVL. עמודים 1-2 במסמך זה מתארים את החלק הזה.
2. חלק ניסויי-תיאורטי: בהתבסס על המימוש מהחלק המעשי, נבצע מספר "ניסויים" עם ניתוח תיאורטי נלווה. עמודים 3-4 מתארים את החלק הזה.

שימו לב: בסוף המסמך (עמוד 4) ישנן הוראות הגשה – הקפידו לפעול לפיהן. תאריך הגשה: 23.5.2023.

חלק מעשי

דרישות

בתרגיל זה יש לממש עץ AVL, לפי ההגדרות שניתנו בכיתה. לכל איבר בעץ יש ערך (info) ומפתח (key) שהוא מספר שלם. כל המפתחות שונים זה מזה, והסדר על צמתי העץ מתייחס כרגיל אך ורק למפתחות.
המימוש יהיה בשפת **python** 3.11 וצריך להיות מבוסס על קובץ השלד המופיע באתר הקורס.
הפעולות שיש לממש הן:

תיאור	פעולה
הפונקציה מחפשת איבר בעל מפתח k. היא מחזירה מצביע לצומת המתאים אם קיים, אחרת מחזירה None.	search(k)
הכנסת איבר בעל ערך s ומפתח k לעץ, ניתן להניח שהמפתח לא קיים כבר בעץ. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ בשלב תיקון העץ על מנת להשלים את הפעולה (גלגולי LR ו- RL נחשבים ל-2 פעולות איזון).	insert(k, s)
מחיקת הצומת x מהעץ בהינתן מצביע. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסך-הכל בשלב תיקון העץ על מנת להשלים את הפעולה.	delete(x)
הפונקציה מחזירה מערך ממוין (ע"פ המפתחות) של האיברים במילון כאשר כל איבר מיוצג ע"י זוג סדור של (key, value).	avl_to_array()
הפונקציה מחזירה את מספר האיברים בעץ.	size()
הפונקציה מקבלת מצביע לצומת x בעץ. עליה להפריד את העץ ל-2 עצי AVL כאשר המפתחות של האחד גדולים מ-x ושל השני קטנים מ-x. יש לממש את הפונקציה על פי המימוש שנלמד בהרצאה. לאחר הקריאה לפונקציה x לא שמיש.	split(x)
הפונקציה מקבלת עץ נוסף t שכל המפתחות שלו קטנים, או שכולם גדולים, מהמפתחות של העץ הנוכחי (שביחס אליו קראנו ל-join) כאשר המפתח k נמצא ביניהם. על הפונקציה לאחד לעץ הנוכחי את העץ הנוסף והאיבר החדש (k, s) כפי שנלמד בהרצאה. על הפעולה להחזיר את ה"עלות" - הפרש גבהי העצים+1. לאחר הקריאה לפעולה העץ t אינו שמיש.	join(t, k, s)
הפונקציה מקבלת מצביע לצומת x בעץ הנוכחי ומחזירה את הדרגה שלו.	rank(x)
הפונקציה מחזירה מצביע לצומת בעל דרגה i, ניתן להניח ש-i דרגה חוקית.	select(i)
הפונקציה מחזירה מצביע לשורש העץ.	get_root()

בנוסף למימוש הפונקציות האלו, יש לממש את מחלקת **AVLNode** כפי שמתואר בקובץ.
מטעמי נוחות, נדרוש שלכל עלה יהיו 2 בנים "וירטואליים", כלומר, צמתים ללא מפתח. באופן זה, נוח יותר לממש גלגולים מכיוון שלכל צומת יהיו 2 בנים.

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

get_key – מחזיר את המפתח של הצומת, או None אם הצומת הוא וירטואלי.
get_value – מחזיר את הinfo של הצומת או None אם הצומת הוא וירטואלי.
get_left – מחזיר את הבן השמאלי של הצומת, או None אם אין כזה.
get_right – מחזיר את הבן הימני של הצומת, או None אם אין כזה.
get_parent – מחזיר את ההורה של הצומת, או None אם אין כזה.

is_real_node – מחזיר TRUE אם הצומת מייצג צומת אמיתי בעץ (קרי: צומת שאינו וירטואלי).
get_height – מחזיר את גובה הצומת, 1- עבור צומת וירטואלי.
get_size – מחזיר את גודל תת-העץ של הצומת, 0 עבור צומת וירטואלי.

<u>הערות חשובות:</u>

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.
3. עליכם לממש את כל הפעולות בסיבוכיות המיטבית.

סיבוכיות

יש לציין בקוד ולהסביר בקצרה במסמך התיעוד את סיבוכיות זמן הריצה במקרה הגרוע (האסימפטוטית, במונחי O הדוקים) של כל פונקציה שמכילה לולאות/רקורסיה, כתלות במספר האיברים בעץ n. עליכם לממש את כל הפעולות בסיבוכיות זמן ריצה טובה ביותר (על פי הנלמד בכיתה).

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 יכלול 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 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 זו נדון ב insertion-sort באמצעות עצי AVL. המיון מתבצע באופן הבא: מכניסים את האיברים לפי הסדר (הלא ממוין) אל העץ, ובסיום מבצעים סריקת in-order לקבלת הסדר הממוין. נגדיר את עלות המיון כסכום עלות החיפושים ומספר פעולות האיזון לאורך המיון.

<u>שימו לב:</u> לצורך שאלה זאת, פעולת החיפוש המתבצעת בהכנסה לעץ AVL (חיפוש מיקום ההכנסה) תמומש בשיטת ה- finger-tree, כלומר החיפוש יתחיל מהאיבר המקסימלי בעץ (ולא מהשורש!).

- לצורך הניתוח, נמיין מערכים בגדלים שונים .גודל המערך שנמיין יהיה $n = 1500 * (2^i)$ כאשר $i =$ 1, ... ,5, ואיבריו יהיו הטבעיים עד n. כלומר, עבור $i = 1$ המערך בגודל 3000, ועבור $i = 5$ המערך בגודל 48,000.
- לכל גודל n של מערך, נבצע **3** ניסויים נפרדים:
  - בניסוי אחד נמיין מערך **ממוין-הפוך**, מגדול לקטן.
  - בניסוי שני סדר האיברים במערך יהיה **אקראי**.
  - **בניסוי שלישי סדר האיברים יהיה כמעט ממוין: ניקח מערך ממוין ונחלק אותו לבלוקים בגודל 300. כעת נהפוך את סדר האיברים בכל בלוק לממוין הפוך.**

1. עבור כל ניסוי, יש לציין את מספר החילופים במערך המקורי, כאשר מספר החילופים מוגדר להיות מספר הזוגות $i > j$ כך ש- $a[i] < a[j]$, ואת עלות המיון הכוללת ב-AVL כפי שהגדרנו מעלה. את המספרים יש למלא בטבלה:
(נזכיר שעלות חיפוש בודד הוא אורך המסלול מהאיבר המקסימלי אל מיקום ההכנסה, כי תצורת העבודה בשאלה היא finger-tree.)

מספר סידורי i	מספר חילופים במערך ממוין-הפוך	עלות מיון AVL עבור מערך ממוין-הפוך	מספר חילופים במערך מסודר אקראית	עלות מיון AVL עבור מערך מסודר אקראי	מספר החילופים במערך כמעט ממוין	עלות מיון AVL עבור מערך כמעט ממוין
1						
2						
3						
4						
5						

<u>הנחייה:</u> כדי לחשב את מספר החילופים ביעילות (בפרט, לעלות מהמקסימום רק עד לאן שצריך ולא למעלה מזה), עליכם לחשב את הדרגה של כל צומת שמכניסים תוך כדי החיפוש שלו. פרטו כיצד מימשתם זאת ואיך חישבתם את מספר החילופים.

1. נתחו באופן תיאורטי את מספר החילופים וגם את עלות החיפושים של ה-AVL במקרה של מערך ממוין-הפוך. התשובות צריכות להיות כתלות בגודל המערך n. את מספר החילופים יש לתת באופן מפורש, עלות החיפושים יכולה להיות מתוארת במונחי $\Theta(\cdot)$ (הציגו חסמי $O(\cdot)$ ו- $\Omega(\cdot)$)

2. האם הערכים בטבלה מסעיף א' והניתוח מסעיף ב' מתאימים? נמקו.
הדרכה: במקרה שמצפים לביטוי ליניארי, ניתן לבדוק את מידת ההתאמה האמפירית של הנתונים על-ידי חישוב קו-מגמה. בתוכנת "אקסל", למשל, ניתן לחשב קו-מגמה ומדד ה-R^2 מעיד על איכות הקירוב. כאשר הקשר אינו לינארי, נעבד תחילה את הנתונים (הפעלת הפונקציה הרלוונטית כך שהקשר כן יהיה לינארי) כדי להעביר אותם לצורה שבה הקשר המצופה יהיה ליניארי (בקואורדינטות החדשות).

שאלה 2:

בשאלה זו נרצה לנתח את העלות של פעולות ה-join המתרחשות במהלך ביצוע split.

- לצורך הניתוח, נבנה עצי AVL בגדלים שונים. מספר איברים שנכניס לעץ יהיה($n = 1500 * (2^i$ כאשר $i = 1, \dots, 10$. כלומר, עבור $i = 1$ העץ בגוד 3000 , ועבור $i = 10$ העץ בגודל כמיליון וחצי. 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 מפתח אקראי בעץ.
  - בניסוי שני נבצע split על המפתח 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 הניסויים (split אקראי או על האיבר המסוים שבחרנו), והסבירו אם התוצאות מתיישבות עם ניתוח הסיבוכיות התאורטי.
מותר להשתמש ללא הוכחה בכך שסיבוכיות פיצול אסימפטוטית היא כעומק הצומת.

3. נתחו באופן תיאורטי את העלות של **join** מקסימלי בניסוי השני של split על האיבר המקסימלי בתת העץ השמאלי של השורש. הסבירו האם התוצאות מתיישבות עם ניתוח הסיבוכיות התאורטי.

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 **AVLTree.py**.
2. קובץ טקסט **info.txt** המכיל את פרטי הזוג באנגלית: מספר ת"ז, שמות, ושמות משתמש.
3. מסמך תיעוד חיצוני, המכיל גם את תוצאות המדידות. את המסמך יש להגיש בפורמט **pdf**.

שמות קובץ התיעוד וקובץ הzip צריכים לכלול את שמות המשתמש האוניברסיטאיים של הזוג המגיש לפי הפורמט **AVLTree_username1_username2**.pdf/zip, בתוכן הקבצים יש לציין את שמות המשתמש, תעודות הזהות ושמות המגישים (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!