

מבני נתונים – פרויקט מספר 1 – עץ מאוזן

הקדמה

בתרגיל זה שני חלקים:

1. חלק מעשי: יימוש של עץ AVL. עמדים 2-1 במסמך מתארים חלק זה.
2. חלק ניסויי-תאורטי: בהתבסס על המימוש מהחלק המעשי, נבצע מספר ניסויים עם ניתוח תאורטי נלווה. עמדים 4-3 מatarsים חלק זה.

שימו לב: בסוף המסמך (עמד 4) ישנן הוראות הגשה – הקפידו לפעול לפיהן. תאריך הגשה: **28/12/2025**

חלק מעשי

דרישות

בתרגיל זה יש למשוך AVL, לפי ההגדרות שניתנו בכיתה. לכל איבר בעץ יש ערך (value) שהוא מחרחת, ומפתח (key) שהוא מספר שלם. כל המפתחות שונים זה מזה, והסדר על המפתח העצמי. כרגע ארכ ווק. למפתחות. המימוש יהיה בשפת **python** 3.13 וצריך להיות מבוסס על קובץ השלד המופיע באתר הקורס. הפעולות שיש למשוך הן:

פעולה	תיאור
search(k)	הfonקציה מחפשת איבר בעל מפתח k. היא מחזירה זוג סדור (e, x), כאשר x הוא מצביע לצומת המתאים (או None אם לא קיים), ו- e הוא אורך בקשנות של מסלול החיפוש+1.
finger_search(k)	הfonקציה מחפשת איבר בעל מפתח k, החל מהתוצמת המקסימלי. היא מחזירה זוג סדור (e, x), כאשר x הוא מצביע לצומת המתאים (או None אם לא קיים), ו- e הוא אורך בקשנות של מסלול החיפוש+1.
insert(k, v)	הכנסת איבר בעל ערך v ומפתח k לעץ, ניתן להניח שהמפתח לא קיים כבר בעץ. הfonקציה מחזירה שלשה (h, e, x), כאשר x מצביע לצומת שנוצר, e מספר הקשנות על מסלול הרכנשה, ו-h מספר מקרי שינוי גובה (promote) שנדרשו במהלך האיזון.
finger_insert(k, v)	הכנסת איבר בעל ערך v ומפתח k לעץ החל מהתוצמת המקסימלי, ניתן להניח שהמפתח לא קיים כבר בעץ. הfonקציה מחזירה שלשה (h, e, x), כאשר x מצביע לצומת שנוצר, e מספר הקשנות על מסלול הרכנשה, ו-h מספר מקרי שינוי גובה (promote) שנדרשו במהלך האיזון.
delete(x)	מחיקת הצומת x מהעץ בהינתן מצביע.
join(t, k, v)	הfonקציה מקבלת עץ t נוסף s שככל המפתחות שלו קטנים ממש, או שכולים גדולים ממש מהמפתחות של העץ הנוכחי, כאשר המפתח k נמצא ביניהם. על הfonקציה לאחד לעץ הנוכחי את העץ הנוסף והאיבר החדש (v, k). לאחר הפעולה העץ t אינו שימושי, יכולות אסור למשתמש (טستر) לקורא לו יותר.
split(x)	הfonקציה מקבלת מצביע לצומת x בעץ. עליה לפרק את העץ לשניים ולהחזיר (t_1, t_2), כך ש- t_1 יכיל את המפתחות הקטנים מ-x ו- t_2 את הגדלים. לאחר הפעולה העץ t אינו שימושי, וגם x אינו שימושי.
avl_to_array()	הfonקציה מחזירה מבירע ממוקין (רשימה של פיטון ממוקנית ע"פ המפתחות) של האיברים במילון כאשר כל איבר מיוצג ע"י זוג סדור של (key, value).
max_node()	הfonקציה מחזירה מצביע לצומת בעל המפתח המקסימלי בעץ.
size()	הfonקציה מחזירה את מספר האיברים בעץ. אין לתזקק שדה size בכל צומת אלא רק של העץ כולו. לאחר ביצוע split מותר שערך לא יהיה תקין, ככלומר לאחר split לא תיבדק size על אף אחד משני העצים שהוחזרו בתוצאות.split()
get_root()	הfonקציה מחזירה מצביע לשורש העץ

לצורך יימוש פעולות אלו, ניעזר במחלקה **AVLNode** המופיעה בקובץ. נדרש לכל עלה יהיו 2 בנימ "וירטוואליים", כלומר צמתים ללא מפתח. באופן זה, נוח יותר למשג גלגולים מכיוון שלכל צומת יהיו 2 בנימ.

המחלקה AVLNode מכילה את השדות הבאים, שעיליכם לתחזק:

- המפתח של הצומת. key
- הערך של הצומת. value
- הבן השמאלי של הצומת. left
- הבן הימני של הצומת. right
- ההורה של הצומת. parent
- גובה הצומת. height

בנוסף, המחלקה תומכת בפעולות הבאה:

- `is_real_node` – מחזירה TRUE אם הצומת מייצג צומת אמיתי בעץ (קרי: צומת שאינו יירטואלי).
- FALSE עבור צומת יירטואלי יש להחזיר.

מספר הבחרות והסבירו:

1. בפעולות החיפוש אנו סופרים את אורךו בקשות של המסלול בעץ בין הצומת שבו התחיל החיפוש והצומת שבו הסתיים החיפוש, ומוסיפים לתוצאה 1.
2. בפעולות הכנסה סופרים את אורךו בקשות של המסלול בעץ בין הצומת שבו התחיל החיפוש והצומת שהכנסנו מיד לאחר הכנסה, לפני פעולות איון. באופן שקול, זהה לאורך המסלול להוראה שלו + 1, וכן אם מchipשים מפתחה של `key` בעץ לפני סמכנים אותו מקבלים `1 = e`.
3. עבור פעולה הכנסה, כאשר מפעלים את אלגוריתם האיזון כפי שנלמד בכיתה, סופרים אך ורק כמה פעמים היינו במקרה 1 (`promote`) של שינוי שדה גובה. אנו מתעלמים מגלגולים ומשינוי גובה שהתרחשו במקרים 2 או 3.
לדוגמה, נתבונן בסדרת הכנסה של 3, ואז 1, ואז 2. בהכנסה הראשונה אין פעולות איון. בהכנסה של 1 מעדכנים את הגובה של 3 (מקרה 1), פעולה אחת. בהכנסה של 2 מעדכנים את הגובה של 1 (מקרה 1) ולאחר מכן בשורש 3 אנחנו נמצאים במקרה 3 של גלגול כפול אותו לא סופרים, لكن שוב מחזירים 1.
4. חיפוש החל מהמקרים: עולים מהמקרים עד הצומת הראשון בעל מפתח קטן יותר (או השורש) וממנו יורדים כרגע. (חיפוש זה ייעיל יותר רק עבור מפתחות גדולות שקבעים למקרים).

הערות חשובות:

1. המימוש יבוצע על ידי מילוי קובץ השלד. מותר להחליף את תוכן הפונקציות המקוריות ולהוסיף פונקציות חדשות. אסור לשנות את חתימות הפונקציות המקוריות ואת שמות השדות המקוריים כדי לא לפגוע ביטטרו (כן מותר להוסיף פרמטרים עם ערך ברירת מחדל). על כל הפונקציות/מחלקות להופיע בקובץ יחיד.
2. אין להשתמש באך מימוש ספרייה של מבנה נתונים.
3. עיליכם למשם את כל הפעולות בסביבות המיטבית.

סיכום

יש לציין בקוד ולהסביר בקצרה במסמך התיעוד את סיבוכיות זמן הריצה במקורה המקורי (האיסימפטוטית, במונחי O הדוקים) של כל פונקציה שמכילה לולאות/רקורסיה, כתלות במספר האיברים בעץ.

פלט

אין צורך בפלט למשתמש.

תיעוד

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

יש להגיש בנוסף לקוד גם מסמך תיעוד חיצוני. המסמך יכול את תיאור המחלקה שモ羞ה, ואת תפקидו של כל חבר במחלקה. עבור כל פונקציה במחלקה יש לפרט מה היא עשויה, כיצד היא פועלת ומה סיבוכיות זמן הריצה שלה. בפרט, אם פונקציה קוראת לפונקציית עזר, יש להתייחס גם לפונקציית העזר בניתו. עבור פונקציות שעולות זמן קבוע יספק תיאור קצר ולא לפרט את ניתוח הסיבוכיות.

בדיקות

התרגילים יידקנו באמצעות תוכנת טسطר שקוראת לפונקציות המפורטות מעלה בתרחישים שונים, ומבודדת את נכונות התוצאות. קובץ הטسطור שלנו לא י포רט לפני הבדיקות. מומלץ מאוד למשתמש אוסף בדיקות עבור המימוש, לא בשליל ההגשה, אלא כדי לבדוק שהקוד לא רק רץ, אלא גם נכון!

בקובץ שתגלו לא תהיה פונקציית **main** ולא יהיו הרצאות קוו/הדפסות, דבר זה יפגע בטسطור שיבדק לכם את התרגילים. אין צורך להגיש את הקוד הנוסף שככבותם חלק הנסיוי.

חלק ניסויי/תאורטי

בשאלה זו נדון ב **insertion sort**-based AVL Finger Tree. המיין מתבצע באופן הבא: מכנים את האיברים לפי הסדר (הלא ממויין) אל העץ, כאשר החיפוש בהכנסת כל איבר חדש מתחילה מהמקסימום היותר, ובסיום מבצעים סריקת **order-in** לקבלת הסדר הממויין. עבור עלות בנייה העץ, ונתח בנפרד את עלות החיפושים ואת מספר פעולות האיזון.

- לצורך ניתוחו, נמיין מערכים בגודלים שונים. גודל המערך שנמיין יהיה $i^2 * 300 = n$ כאשר $i = 10, \dots, 1, \dots, 10$, ואיברו יהיו הטבעיים עד n . למשל, עבור $i = 1$ המערך בגודל 300, ועבור $i = 5$ המערך בגודל 9600.
- לכל גודל n של מערך, נבצע 4 ניסויים נפרדים:
 - בניסוי הראשון נמיין מערך ממויין, מקטן לגודל.
 - בניסוי השני נמיין מערך ממויין הפוך, מגודל לקטן.
 - בניסוי השלישי סדר האיברים במערך יהיה אקריאי.
 - בניסוי הרביעי ניקח מערך ממויין ובעור כל אינדקס פרט לאחרון $2 - i, \dots, 0$, נבצע החלפה עם האיבר הבא $A[i + 1] \leftrightarrow A[i]$ בסיסי חצי (שימוש לב שייתכן שאיבר יוחלף מספר פעמים).

הערה: בסעיפים הבאים, עבור ניסויים אקריאים, יש לחת את הממוצע על פני 20 ניסויים.

1. יש למלא בטבלה הבאה את סך עליות האיזון **לא בגלולים** עבור כל אחד מהניסויים. הסבירו מהו החסם העליון התאורטי על סך עליות האיזון כולל גלגולים, והאם הערכים בטבלה מתאימים. לסימן, נמקו מדוע תוספת הגלגולים לשפירה אינה אסימפטוטית.

מספר סידורי ?	מספר ממויין	עלות איזון במערך	מערך ממויין-הפוך	עלות איזון במערך מסודר אקראית	עלות איזון במערך היפוכים סמווכים אקריאים
1					
2					
...					

2. בהינתן מערך A בגודל n נגדיר היפוך בתווך זוג אינדקסים $j < i$ כר' שמתקיים $[j < i] < A[i] < [i < j]$, ונסמן את מספר היפוכים הכלול במערך ב- I . ניתן לשים לב כי באופן כללי ($\binom{n}{2} \leq I \leq 0$ וככל שיש פחות היפוכים כך המערך קרוב יותר לממיין. יש למלא בטבלה הבאה את מספר היפוכים במערך הקליט עבור כל אחד מהניסויים (מספיק עד 5).

מספר סידורי ?	מספר ממויין	מספר היפוכים במערך ממויין-הפוך	מספר היפוכים במערך מסודר אקראית	מספר היפוכים במערך	מספר היפוכים סמווכים אקריאים
1					
2					
...					

3. יש למלא בטבלה הבאה את סך עלויות החיפוש עבור כל אחד מהניסיונות.

מספר סידורי	עלות חיפוש במערך ממון	עלות חיפוש במערך מנויין-הפון	עלות חיפוש במערך מסודר אקראית	עלות חיפוש במערך עם היפוכים סמוכים אקראיים
1				
2				
...				

4. כדי לחסום מלמעלה את סך עלויות החיפוש באופן תאריטי, נסמן ב- d_i את מספר האיברים לפניו האיבר באינדקס i שגדולים ממנו.

$$\text{I. } \text{הסבירו מדויק } d_i = I.$$

II. חסמו את עלות החיפוש בעת הכנסת האיבר $-i$ כפונקציה של d_i , והסבירו כי סך עלויות החיפוש לאוריך סדרת הנקודות הוא $(2 + \sum_{i=1}^n \log(d_i))$.

הערה: כאשר הסיבוכיות היא $O(\max(1, \log d))$ עבור $0 \leq d \leq n$, ניתן להמיר לביטוי $(2 + \log(d))$ כדי לעבור לביטוי אחד פשוט וחוקי שיחסם את שניהם מלמעלה.

III. השתמשו באי-שוויון המוצעים כדי לחסום מלמעלה את סך עלויות החיפוש כפונקציה של I , *a*. הסבירו מדויק זהה הערות הדומיננטיות מבחן אסימפטוטית.

IV. השוו בין החסם שהתקבל ותוצאות הניסויים.

הוראות הגשה

הגשת התרגיל תבוצע באופן אלקטרוני באתר הקורס במודול.

הגשת התרגיל היא בזוגות בלבד!

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

על ההגשה לכלול שלושה קבצים:

1. קובץ המקור (הרחבה של קובץ השלד המקורי) תחת השם **AVLTree.py**.

2. קובץ טקסט **info.txt** המכיל את פרטי הזוג: שמות מלאים בעברית, שמות מלאים באנגלית ומספר ת"ז. יש לרשום גם שמות משתמש (או לחילופין אימיילים או ניבראטיבאים) של המתגים.

3. מסמך תיעוד חיצוני, המכיל גם את תוכנות המדידות. את המסמך יש להגיש בפורמט **pdf**.

שמות קובץ התיעוד וקובץ הקובץ צריכים לכלול את שמות המשתמש האוניברסיטאיים של הזוג הנגיש לפיה הפורטט **zip** **username1_username2.pdf**, בתוכן הקבצים יש לציין את שמות המשתמש, תעודות הזהות ושמות המתגים (בכותרת המסמך ובשורת הערה בקובץ המקורי).

הגשת שיעורי הבית באיחור - באישור מראש בלבד. הגשה באיחור ללא אישור תגרור הורדת נקודות מהציון.

הגשת התרגיל היא חובה לשם קבלת ציון בקורס.

בצלחה!