אוניברסיטת תל אביב

סמסטר א' תשפ"ג

**מבני נתונים - פרויקט חקר תכנותי מספר 1 - עץ דרגות**

**הקדמה:**

בתרגיל זה שני חלקים:

1. חלק המעשי: מימוש של List באמצעות 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 ושאלות חקר. עמודים 3-5 מתארים את החלק הזה.

**שימו לב:** בסוף המסמך (עמוד 5) ישנן הוראות הגשה – הקפידו לפעול לפיהן.

**תאריך הגשה:** 30/12 .

בנוסף, יש לעקוב אחר השרשור הנעוץ 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 . |
| 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. |
| permutation() | הפונקציה מחזירה את אותם האיברים ברשימה בסדר אקראי |
| sort() | הפונקציה מסדרת את איברי הרשימה בסדר עולה. |
| concat(lst) | הפונקציה מקבלת רשימה. על הפונקציה לשרשר אותה אל סוף הרשימה הנוכחית. על הפעולה לרוץ בזמן . יש להחזיר את הערך המוחלט של הפרש הגבהים של עצי הAVL שמוזגו. |
| search(val) | החזרת האינדקס הראשון ברשימה בו מופיע הערך val, או 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 אם הצומת הוא וירטואלי.

getValue – מחזיר את הinfo 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 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 להביא רק תיאור קצר ולא לפרט את ניתוח הסיבוכיות.

מחלקת :

מחלקה זו מכילה את השדות הבאים:

* - ערך הצומת.
* - מצביע לבן השמאלי, מאותחל בתור צומת וירטואלי, (מפורט בהמשך).
* - מצביע לבן הימני, מאותחל בתור צומת וירטואלי, .
* - מצביע לאב הצומת, מאותחל בתור .
* - גובה הצומת (גובה של עלה מוגדר להיות , של צומת וירטואלי ).
* - מספר הצמתים בתת העץ כולל הצומת הנוכחי.

מחלקה זו מכילה את הפעולות הבאות אשר **כולן בסיבוכיות**  ומשתמשות בפונקציית לבדיקה אם הצומת היא וירטואלית או אמיתית:

* - מחזירה את שדה של הצומת, או אם אין כזה.
* - מחזירה את שדה של הצומת, או אם אין כזה.
* - מחזירה את השדה של הצומת, או אם אין כזה.
* - מחזירה את שדה של הצומת או אם הצומת הוא וירטואלי.
* - מחזירה את שדה של הצומת. עבור צומת וירטואלי הגובה .
* - מחזירה את שדה של הצומת. עבור צומת וירטואלי .
* – מחזירה את הbalance factor של הצומת. חישוב: גובה הבן השמאלי פחות גובה הבן הימני.
* - מגדירה את השדה להיות ה שהוכנס.
* - מגדירה את השדה להיות ה שהוכנס.
* - מגדירה את השדה להיות ה שהוכנס.
* – מגדירה את השדה להיות ה שהוכנס ואת שדה של ה שהוכנס בקלט להיות האיבר עצמו.
* - מגדירה את השדה להיות ה שהוכנס ואת שדה של ה שהוכנס בקלט להיות האיבר עצמו.
* – מגדירה את שדה להיות ה של הקלט.
* - מגדירה את שדה של הצומת לפי הערך שהוכנס.
* - מגדירה את שדה של הצומת. מאותחלת עם הערך 0.
* - מחזירה אם הצומת מייצג צומת אמיתי בעץ. כלומר צומת שאינו וירטואלי וזאת כאשר .
* – מעדכן את שדה של ה בחישוב ה של הבנים שלו כפי שנלמד בכיתה.
* – מעדכן את שדה של ה בחישוב ה של הבנים שלו כפי שנלמד בכיתה.

מחלקת :

מחלקה זו מכילה את השדות הבאים:

* - מחזיק את כמות הצמתים בעץ.
* - מצביע על שורש העץ, על איבר מסוג .
* – מצביע לאיבר הראשון ברשימת האיברים הממוינת בסדר עולה.
* – מצביע לאיבר האחרון ברשימת האיברים הממוינת בסדר עולה.

מחלקה זו מכילה את הפעולות הבאות:

* - פעולה אשר מחזירה אם שורש העץ אינו צומת אמיתי, כלומר העץ ריק, ו- אחרת. פעולה זו בסיבוכיות שכן היא פונה לשדה של העץ ועובדת כמו פונקציית שעובדת בסיבוכיות זו.
* – מחזירה את ערכו של הצומת במיקום ה בגודלו ברשימה. פונקציה זו משתמשת במתודת המפורט תחת פונקציות העזר, אשר בסיבוכיות ועל כן זו הסיבוכיות של מתודה זו.
* – מכניסה את הערך בקלט למיקום ברשימה במידה שקיימים לפחות איברים ברשימה. המתודה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ לטובת תיקון העץ. במתודה זו אנו משתמשים במתודות עזר אשר מחלקות למקרים כל סוג הכנסה.

מתודות עזר:

* – מבצעת את פעולת כפי שנלמד בכיתה. משתמש בפונקציית ולכן סיבוכיותה הינה כדי למצוא את האיבר ה בדרגתו.
* – פונקציה אשר מחזירה את הצומת הראשון אשר גודלו הינו לכל הפחות עבור אי-שלילי שקטן ממספר האיברים במבנה. מכיוון שיש ברשות המבנה שלנו מצביע למינימום (גישה אליו מיידית), מדובר בטיול מהמינימום כלפי מעלה בלבד עד אשר מוצאים את הצומת המתאים, שהוא במקרה הכי גרוע השורש ולכן הסיבוכיות תוך התקדמות בשדרת האבות מהמינימום לשורש העץ הינה כפי שלמדנו.
* – מימוש של כפי שנלמד בכיתה, כולל רקורסיבי. כמו כן למדנו כי מיון זה הינו במקרה הגרוע .
* – מימוש אלגוריתם אשר מבצע סידור רנדומי למערך באופן שהוא . אנו רצים בלולאה במשך מספר האיברים ובכל פעם בוחרים רנדומית אינדקס (בטווח שהולך ויורד בכל איטרציה), ואז מבצעים החלפה לזוג איברים. נבחין כי אנו מניחים כי השימוש בפונקציית מספריית הינו לכל היותר קבוע ולכן מדובר בסיבוכיות במקרה הגרוע.
* - מחזירה את השדה של העץ , אלא אם מדובר בצומת וירטואלי עבורו היא מחזירה את הערך . גישה לשדה בסיבוכיות .
* - פעולה המחזירה את ערך של הצומת בעלת המפתח , אם קיימת. פעולה זו משתמשת בפונקציה בסיבוכיות ולכן זוהי גם סיבוכיותה של פונקציית .
* - פונקציה אשר מקבלת צומת וערך מפתח ובודקת על-ידי "טיול" על גבי העץ מהשורש ועד לעלה העץ אם הגענו בהתאם לתכונות של עץ חיפוש בינארי לערך המפתח המבוקש. כלומר התקדמות לעומק העץ מתבצעת לפי הלוגיקה של אם ערך המפתח גדול מערך הצומת אותה אנו מבקרים, נמשיך לטייל ימינה לערכים גדולים יותר ושמאלה אחרת. אם הגענו לעלה וירטואלי סימן שלא מצאנו את הצומת המבוקשת כי היא אינה קיימת ונחזיר . מכיוון שעומק עץ הינו וכל הפעולות שאנו מבצעים הינן פעולות בזמן קבוע הרי שמדובר בסיבוכיות לרקורסיה כולה.
* - תחילה נחפש את הצומת באמצעות פעולת ואם היא כן נמצאת נחזיר שכן אסור ב להחזיק שני צמתים עם אותו . אחרת, נאתחל משתנה הבנוי מ שהגיעו כקלט. נבדוק אם הצומת החדש הוא המקסימום או המינימום החדש בעץ ואם כן נעדכן את השדות המתאימים. אם העץ ריק, נבדוק באמצעות שימוש בפונקציית , נעדכן את השדות המתאימים ונחזיר 0 (כמספר פעולות קידום/סיבוב). נגדיר משתנה שמתעד את מספר פעולות הקידום והסיבוב. באמצעות פונקציית , בעלת סיבוכיות נמצא את אביו של הצומת המיועד להכנסה ונכניסו לצד המתאים של אביו באמצעות ערכו ונעדכן את השדות המתאימים. כעת נבדוק אם הכנסנו את הצומת כבן שמאלי/ימני של צומת אונארי או של עלה. בהתאם, נעדכן את השדות המתאימים עד השורש. נבחין כי אם הכנסנו את הצומת לעלה, אז עלינו לבצע את פעולת בסיבוכיות של כדי לאזן את העץ מחדש. תוך כדי הפעולה נשמור את מספר הקידומים/סיבובים שבוצעו ונחזיר ערך זה. נבחין כי כל הפעולות הינן בלתי תלויות ולא מקוננות ולכן בסך הכל מדובר בסיבוכיות של .
* - פונקציה איטרטיבית אשר תכליתה למצוא משורש העץ על-פי כללי החיפוש את הצומת עבורה . אם זה לא קיים בעץ, אזי נחזיר את הצומת שהוא הכי קרוב מבחינת תכונות החיפוש ל כפי שנלמד בהרצאה. נבחין כי במקרה הגרוע אנו נבצע ירידה אחת עד לעלה וירטואלי כלשהו ולכן סיבוכיות פעולה זו הינה .
* - פונקציה אשר מקבלת שני צמתים, אב ובן. מבצעת החל מהבן ועד לשורש העץ פעולות איזון בהתאם לנלמד בהרצאה, וסופרת באמצעות את מספר פעולות הקידום/סיבוב שבוצעו במהלך הטיול. במידה ויש לבצע סיבוב או סיבוב כפול, נעזר בפעולות למיניהן. לאחר פעולות האיזון נעדכן את שורש העץ במידת הצורך, ונחזיר את מספר פעולות הקידום/סיבוב אשר התבצעו במשך פונקציה זו. במידה ולא הגענו לשורש העץ מפעולות איזון (כלומר העץ התאזן לפני כן), נמשיך לטייל עד שורש העץ כדי לעדכן את השדות המתאימים. מכיוון שאנו מטיילים רק במעלה העץ, במקרה הגרוע נטייל מעלה ועד לשורש וכן נבחין כי הסיבוכיות של פונקציות העזר הינן , הסיבוכיות כולה הינה .
* - פונקציה אשר מבצעת סיבוב שמאלה כפי שנלמד בהרצאה בהינתן שני צמתים, אב ובן. סיבוכיות .
* - פונקציה אשר מבצעת סיבוב ימינה כפי שנלמד בהרצאה בהינתן שני צמתים, אב ובן. סיבוכיות .
* - פונקציה אשר מבצעת סיבוב כפול, ימין ואז שמאל, כפי שנלמד בהרצאה בהינתן שלושה צמתים, אב, בן ונכד. סיבוכיות .
* - פונקציה אשר מבצעת סיבוב כפול, שמאל ואז ימין, כפי שנלמד בהרצאה בהינתן שלושה צמתים, אב, בן ונכד. סיבוכיות .
* - תחילה נבדוק שהאיבר נמצא בעץ. נמצא את האיבר שצריך למחוק באמצעות קריאה ל- ונכנו . אם גודל העץ הוא אזי אנו מוחקים את השורש ולכן נאפס את העץ ונסיים. אם לצומת שאנחנו רוצים למחוק יש שני ילדים, נמצא את ה שלה, נמחק את הצומת ונשים במקומו את ה שלו. במקרה זה נקרא לפונקציית ה- ונאזן את מעלה העץ החל ממיקום איבר ה- אחרי המחיקה. במקרה קצה שבו האיבר שנמחק הינו האבא של ה-, נבודד נקודתית את . אם הינו עלה, פשוט נמחק את העלה. במקרה האחרון, הינו צומת אונארי ולכן נעקוף את . נפחית את גודל של העץ באחד, ותוך כדי נעדכן את השדות המתאימים (גבהים וגדלים) של האיברים המתאימים. נאתחל משתנה אשר מונה את מספר הקידומים והסיבובים שנעשים תוך כדי האיזון. נטייל החל מאביו של המיקום של הצומת שמחקנו ועד לשורש ונבצע פעולות איזון בהתאם לנלמד בכיתה. לבסוף, נבדוק אם מחקנו את האיבר המינימלי או המקסימלי של העץ ואם כן נעדכן את השדות בהתאמה על-ידי שימוש ב של המינימלי או למקסימלי. ונחזיר את משתנה . כפי שהוכח בכיתה סיבוכיות פעולה זו הינה ב. נבחין כי השימוש בפונקציות חיצוניות הן בלתי תלויות ואינן גוברות על סיבוכיות זו, כלומר כולן לכל היותר בסיבוכיות .
* - פונקציה איטרטיבית אשר מטיילת משורש העץ ובכל פעם מתקדמת ימינה עד שהיא מגיעה לעלה שהוא הצומת בעל המפתח המקסימלי. סיבוכיותה כעומק העץ .
* - פונקציה איטרטיבית אשר מטיילת משורש העץ ובכל פעם מתקדמת שמאלה עד שהיא מגיעה לעלה שהוא הצומת בעל המפתח המינימלי. סיבוכיותה כעומק העץ .
* - פונקציה אשר מקבלת צומת. אם יש לו בן ימני היא תחזיר את הצומת הקטן ביותר של תת עץ ימני זה. אחרת, היא תעלה עד השורש עד שנעלה בפעם הראשונה מצד ימין ונחזיר צומת זה. אם מדובר באיבר הכי גדול נחזיר . סיבוכיותה כעומק העץ, שכן במקרה הגרוע אנו נטייל מעלה כלשהו ועד שורש העץ.
* - מחזירה את השדה של העץ . גישה לשדה בסיבוכיות .
* - מחזירה את השדה של העץ . גישה לשדה בסיבוכיות .
* - נאתחל מערך בגודל העץ. נמצא את המינימום של העץ באמצעות ונכניס את המפתח הראשון. נבצע פעולות ובכל פעם נכניס את המפתח המתאים למערך. סיבוכיות הפעולה, כפי שנלמד בתרגול, הינו . כל פעולות העזר שהשתמשנו הינן מסיבוכיות של אולם כפי שנלמד בתרגול הסיבוכיות הכוללת הינה .
* - נאתחל מערך בגודל העץ. נמצא את המינימום של העץ באמצעות ונכניס את המפתח הראשון. נבצע פעולות ובכל פעם נכניס את ערך הצומת המתאים למערך. סיבוכיות הפעולה, כפי שנלמד בתרגול, הינו . כל פעולות העזר שהשתמשנו הינן מסיבוכיות של אולם כפי שנלמד בתרגול הסיבוכיות הכוללת הינה .
* - מחזירה את השדה של העץ . גישה לשדה בסיבוכיות .
* - נבדוק תחילה אם שני העצים ריקים. אם כן, נחזיר עץ שבו רק הצומת המקשרת נמצאת. לאחר מכן נבדוק אם אחד העצים ריקים, אם כן, נוסיף את הצומת לעץ שאינו ריק, נבצע פעולות איזון באמצעות ונחזיר את גובה העץ שאינו ריק ועוד 1. אחרת, נבחין תחילה כי ישנם 4 מקרים זרים ומשלימים: העץ הנוכחי גבוה מהעץ הנוסף ועם ערכים גדולים יותר ממנו, העץ הנוכחי גבוה מהעץ הנוסף ועם ערכים קטנים יותר ממנו, העץ הנוכחי נמוך מהעץ הנוסף ועם ערכים גדולים ממנו, העץ הנוכחי נמוך מהעץ הנוסף ועם ערכים קטנים ממנו. כל המקרים סימטריים זה לזה ובה"כ נניח כי העץ הנוכחי גבוה יותר ובעל ערכים גדולים יותר מהעץ הנוסף. נטייל משורש העץ הנוכחי ועד שנגיע לצומת הראשונה שהיא בגובה זהה לגובה העץ הנוסף או אחד מעל. נחבר את הצומת המקשרת לשני העצים, נבדוק מקרה מיוחד שבו שני העצים באותו גובה ולכן אין לצומת המקשרת אבא. נעדכן את שדות (במקרים הזרים והמשלימים מעדכנים שדות אחרים בהתאם לאיזה עץ יש ערכים גדולים יותר). נבצע במידת הצורך פעולות איזון (כולל המקרה המיוחד ל- שהוסבר בהרצאה) על-ידי טיול מאביו של הצומת המקשרת (אם היא קיימת) ועד השורש של העץ הנוכחי. נעדכן את כל השדות המתאימים לכל השדות בזמן שמטיילים במעלה השורש, ונחזיר את הפרשי הגבהים בין העצים ועוד 1. סיבוכיות פעולה זו הינו כי במקרה הגרוע נעלה מעלה העץ ועד השורש וכן השימוש בפונקציות העזר הינו לכל היותר .

- נמצא את הצומת המפריד באמצעות . נגדיר שני עצים חדשים שהשורש שלו הוא תת העץ השמאלי של הצומת המפריד, שהשורש שלו הוא תת העץ הימני של הצומת המפריד. נטייל מהצומת המפריד עד השורש. אם עלינו מצד ימין, נאחד באמצעות פעולת את הצומת שאליה עלינו, ואת תת העץ השמאלי של אותה צומת, לעץ . אחרת, עלינו משמאל ולכן נאחד באמצעות פעולת את הצומת שאליה עלינו, ואת תת העץ הימני של אותה צומת, לעץ . לאחר סיום הטיול, נעדכן את שדות לכל תת עץ באמצעות שימוש ב וכן נעדכן את שדה . נבצע השמה של שני העצים לרשימת עצים בגודל 2 כאשר עבורה האיבר במקום ה- הינו ובמקום ה- הינו ונחזיר את רשימה זו. במהלך הפעולה טיילנו במקרה הגרוע מהעלה ועד השורש ולמדנו בכיתה שלמרות פעולות ה- בדרך הסיבוכיות היא והשתמשנו לבסוף בפעולות החסומות ב אשר הינן בלתי תלויות ובסך הכל הסיבוכיות הינה .

**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 מעורבבת של הכנסות ומחיקות.

* לצורך הניתוח, נבנה עצי AVL בגדלים שונים. מספר איברים שנכניס לעץ יהיה כאשר i=1,…,10. כלומר, עבור i=1 העץ בגודל 3000, ועבור i=10 העץ בגודל כמיליון וחצי.
* 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 |  |  |  |

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

**שאלה 2:**

בניסוי זה נשווה את ביצועי עץ AVL לעומת רשימה מקושרת ומערך.

ממשו רשימה באמצעות רשימה מקושרת ובאמצעות מערך (אין צורך להגיש את המימוש הזה).

עבור כל חזרו על התהליך הבא:

* הכניסו איברים לרשימה הממומשת בעזרת עץ AVL (לפי שלושת המקרים המפורטים בהמשך).
* הכניסו את אותם האיברים לרשימה הממומשת בעזרת רשימה מקושרת ומערך.
* עבור כל אחד מהעצים, מדדו את זמן הריצה הממוצע (הממוצע על פני ההכנסות).

חזרו על התהליך כאשר ההכנסות הן לפי:

1. **הכנסות להתחלה** – סדרת הכנסות לראש הרשימה.
2. **סדרה אקראית** - מכניסים במקומות אקראיים מתוך גודל הרשימה הנוכחי.
3. הכנסות לסוף – סדרת הכנסות לסוף הרשימה

רשמו את התוצאות בטבלאות הבאות:

|  |  |  |  |
| --- | --- | --- | --- |
| זמן ריצה בממוצע  מספר סידורי | עץ AVL  הכנסות להתחלה | רשימה מקושרת  הכנסות להתחלה | מערך הכנסות להתחלה |
| 1 |  |  |  |
| 2 |  |  |  |
| ... |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| זמן ריצה בממוצע  מספר סידורי | עץ AVL  הכנסות אקראיות | רשימה מקושרת הכנסות אקראיות | מערך הכנסות אקראיות |
| 1 |  |  |  |
| 2 |  |  |  |
| ... |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| זמן ריצה בממוצע  מספר סידורי | עץ AVL  הכנסות בסוף | רשימה מקושרת הכנסות בסוף | מערך הכנסות בסוף |
| 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 הzip 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 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!**