

## מבני נתונים - פרויקט מספר 1 - עץ דרגות

### תיאור הבעיה

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

1. empty() - פונקציה בוליאנית שמחזירה ערך TRUE אם ורק אם העץ ריק.
2. search(int k) - הפונקציה מחפשת בעץ איבר בעל המפתח k. אם קיים איבר כזה, היא מחזירה את הערך השמור עבורו, אחרת היא מחזירה null.
3. insert(int k, boolean s) - הכנסת איבר בעל ערך s ומפתח k לעץ, אם הוא לא קיים כבר. הפונקציה מחזירה את מספר הצמתים בהם נדרשה פעולת איזון (רוטציה/שינוי גובה) בשלב האיזון שלאחר ההכנסה (כולל הצומת המוכנס בעצמו). אם קיים איבר בעל מפתח k בעץ הפונקציה מחזירה 1- ולא מתבצעת הכנסה.
4. delete(int k) - מחיקת איבר בעל המפתח k בעץ, אם הוא קיים. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ בשלב תיקון העץ על מנת להשלים את הפעולה. אם לא קיים איבר בעל המפתח k בעץ הפונקציה מחזירה 1-.
5. min() - מחזירה את ערכו (info) של האיבר בעץ בעל המפתח המינימלי, או null אם העץ ריק.
6. max() - מחזירה את ערכו (info) של האיבר בעץ בעל המפתח המקסימלי, או null אם העץ ריק.
7. keysToArray() - הפונקציה מחזירה מערך ממין המכיל את כל המפתחות בעץ, או מערך ריק אם העץ ריק.
8. infoToArray() - הפונקציה מחזירה מערך בוליאנים המכיל את כל הערכים בעץ, ממוינים על פי סדר המפתחות. כלומר, הערך ה j במערך הוא הערך הבוליאני המתאים למפתח שיופיע במיקום ה j במערך הפלט של הפונקציה keysToArray(). גם הפונקציה הזאת מחזירה מערך ריק אם העץ ריק.
9. size() - הפונקציה מחזירה את מספר האיברים (צמתים) בעץ.
10. getRoot() - מחזיר את השורש של העץ (אובייקט AVLNode)
11. successor(AVLNode node) - מקבלת צומת בעץ כקלט ומחזירה את העוקב שלו (אם לא קיים עוקב, מחזירה null)
12. prefixXor(int k) - הפונקציה מקבלת מפתח k כאשר נתון ש-k נמצא במבנה. הפונקציה מחזירה את ה-Xor (Exclusive Or) של הערכים הבוליאניים הנמצאים במבנה תחת מפתחות שקטנים או שווים ל-k. תזכורת: אם סט ערכים בוליאנים מכיל מספר אי-זוגי של True, ה-Xor שלהם הוא True, ואחרת הוא False. על המימוש של פעולה זו לקחת זמן לוגריתמי, כלומר  $O(\log n)$  כאשר n מספר הצמתים בעץ.

13. `succPrefixXor(int k)` – מימוש יותר נאיבי של `prefixXor`. מימוש זה צריך להתחיל מהצומת עם המפתח הקטן בעץ, ולבצע פעולות `successor` עד שיגיע לצומת בעל המפתח  $k$ , ואז להוציא כפלט את ה-`xor` של אותם הצמתים עליהם עבר.

**הערה טכנית לגבי ג'אווה:** שימו לב להבדל בין `boolean` לבין `Boolean`. הראשון הוא טיפוס פרימיטיבי שמקבל רק `true` ו-`false`. השני הוא `wrapper`, כלומר מצביע לטיפוס פרימיטיבי. ככזה, הוא יכול לקבל את הערך `null`.

בנוסף למימוש הפונקציות האלו, יש לממש את מחלקת `AVLNode` כפי שמתואר בקובץ. מטעמי נוחות (יקל עליכם לממש גלגולים מכיוון שלכל צומת יהיו 2 בנים), נדרוש שלכל עלה יהיו 2 בנים "וירטואליים", כלומר, צמתים ללא מפתח. ראו שקפים 16-17 במצגת על עצי חיפוש בינאריים.

ל `AVLNode` יש את הפונקציות הבאות (את המפרט המלא תמצאו בקובץ השלד):

- `getKey` – מחזיר את המפתח של הצומת, או -1 אם הצומת הוא וירטואלי
- `getValue` – מחזיר את ה-`info` של הצומת או `null` אם הצומת הוא וירטואלי
- `getLeft` – מחזיר את הבן השמאלי של הצומת, או `null` אם אין כזה
- `getRight` – מחזיר את הבן הימני של הצומת, או `null` אם אין כזה
- `isRealNode` – מחזיר `True` אם הצומת מייצג צומת אמיתי בעץ (צומת שאינו וירטואלי), אחרת `False`
- `getHeight` – מחזיר את גובה הצומת (1- עבור צומת וירטואלי).  
את כל פעולות `AVLNode` יש לבצע ב- $O(1)$ .

בקובץ השלד מופיעים ה-`header`-ים של כל הפונקציות. 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. כמובן, עליכם לממש את ה-`l` בעצמכם, ללא שימוש בשום ספרייה של מבני נתונים.

## סיבוכיות

יש לתעד בקוד ובמסמך נפרד (ביותר פירוט) את סיבוכיות זמן הריצה במקרה הגרוע (האסימפטוטית, במונחי  $O$  הדוקים) של כל פונקציה, כתלות במספר האיברים בעץ  $n$ . עליכם להשיג סיבוכיות זמן ריצה (במקרה הגרוע ביותר) נמוכה ככל הניתן עבור כל אחת מהפונקציות, והיכן שמצויינת דרישת סיבוכיות עליכם כמובן לעמוד בה. במסמך הנפרד עליכם להצידיק סיבוכיות של פעולות אם הניתוח איננו טריוויאלי.

## פלט

אין צורך בפלט למשתמש.

## תיעוד

בנוסף לבדיקות אוטומטיות של הקוד שלכם, 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, על אורך השורות, מבנה פשוט לקוד וכו'. ככתוב מעלה, יש להגיש בנוסף לקוד גם מסמך תיעוד חיצוני. המסמך יכלול את תיאור המחלקה שמומשה, ואת תפקידו של כל חבר במחלקה. עבור כל מתודה 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.

## בדיקות

התרגילים ייבדקו באמצעות תוכנת טסטר שקוראת לפונקציות המפורטות מעלה, ומוודאת את נכונות התוצאות. קובץ הטסטר שלנו **לא יפורסם** לפני הבדיקות. עליכם לבדוק את המימוש בעצמכם! בפרט, כדאי מאוד לממש טסטר, כדי לבדוק את תקינות ונכונות המימוש, כולל במקרי קצה שונים.

**חשוב:** בקובץ שתגישו לא תהיה פונקציית main (דבר זה יפגע בטסטר שיבדוק לכם את התרגילים). אם הצלחתם לקמפל את הפרויקט לבדו (ללא טסטר), זה סימן שמשהו לא נכון במימוש שלכם.

הקוד ייבדק על מחשבי בית הספר על גירסא Java8.

הנחיות להשמת סביבת העבודה בבית (ג'אוה+אקליפס):

<http://courses.cs.tau.ac.il/software1/1415b/misc/workenv.pdf>

מדריך לעבודה עם Eclipse (סעיפים 9-15):

<http://www.vogella.com/>

הנחיות לפתיחת חשבון מחשב, למי שמעוניין/ת לעבוד במעבדת בית הספר:

<http://cs.tau.ac.il/system/accounts0>

שימוש בג'אוה 8 במעבדות האוניברסיטה:

<http://courses.cs.tau.ac.il/software1/1415b/misc/lab-eclipse.pdf>

## מדידות

(1) בניסוי זה נשווה את הפונקציה היעילה prefixXor ואת הפונקציה הנאיבית succPrefixXor.

חזרו על התהליך הבא עבור כל  $i=1, \dots, 5$ :

1. הכניסו לעץ AVL  $n = 500 * i$  איברים טבעיים אקראיים שונים.
2. עברו על המפתחות שהכנסתם בסדר עולה, וקראו ל- $\text{prefixXor}(k)$  עבור כל מפתח  $k$  שאתם עוברים עליו. רשמו בטבלה את:
  - הזמן הממוצע עבור קריאה ל- $\text{prefixXor}$  על פני הקריאות שביצעתם.
  - הזמן הממוצע עבור קריאה ל- $\text{prefixXor}$  מתוך 100 הקריאות הראשונות שביצעתם (כלומר, על 100 המפתחות המינימליים במבנה).
3. חזרו על הסעיף הקודם עם succPrefixXor במקום prefixXor.

מספר סידורי	עלות prefixXor ממוצעת (כל הקריאות)	עלות succPrefixXor ממוצעת (כל הקריאות)	עלות prefixXor ממוצעת (100 קריאות ראשונות)	עלות succPrefixXor ממוצעת (100 קריאות ראשונות)
1				
2				
...				

רשמו את התוצאות בטבלה כמו זו שלמטה, והסיקו מסקנות לגבי זמן הריצה של הפעולות. האם התוצאות מתיישבות עם הניתוח התאורטי של סיבוכיות הזמן? הסבירו.

(2) בניסוי זה נשווה את ביצועי עץ AVL עם עץ חיפוש בינארי רגיל (לא מאוזן).

ממשו מחלקה המייצגת עץ לא מאוזן שכזה, למשל ע"י הסרת הגלגולים מהמימוש הקיים לעץ 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  $i \in \{1, \dots, 5\}$  חזרו על התהליך הבא:

- הכניסו  $n = 1000 * i$  איברים לעץ AVL (לפי שלושת המקרים המפורטים בהמשך).
- הכניסו את אותם האיברים לעץ חיפוש לא מאוזן.
- עבור כל אחד מהעצים, רשמו את הזמן הממוצע פר הכנסה בודדת (הממוצע על פני  $n$  ההכנסות).

חזרו על התהליך כאשר  $i = 1000$  האיברים הם:

1. **סדרה חשבונית** - כלומר המספרים  $1, 2, 3, \dots, 1000 * i$
2. **סדרה מאוזנת** - סדרת מספרים המניבה עץ בעל הגובה המינימלי האפשרי גם ללא מנגנון איזון – כלומר שגובה העץ הוא  $\lceil \log n \rceil$  (כאשר  $n = 1000 * i$ ). עליכם לחשוב על סדרת מספרים שגורמת לכך.
3. **סדרה אקראית** - מספרים טבעיים אקראיים.

רשמו את התוצאות בטבלה הבאה:

עלות הכנסה ממוצעת	מספר סידורי $i$	עץ ללא מנגנון איזון סדרה חשבונית	עץ ללא מנגנון איזון סדרה מאוזנת	עץ ללא מנגנון איזון סדרה אקראית	עץ AVL סדרה חשבונית	עץ AVL סדרה מאוזנת	עץ AVL סדרה אקראית
	1						
	2						
	...						

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

### הגשה

הגשת התרגיל תתבצע באופן אלקטרוני באתר הקורס במודל.

**הגשת התרגיל היא בזוגות בלבד!** על המעוניינים לחרוג מכך (להגיש ביחיד, או במקרים מוצדקים בשלישיה) לפנות באופן אישי למתרגלים.

כל זוג ייבחר נציג **אחד** ויעלה תחת שם המשתמש שלו את קבצי התרגיל (תחת קובץ zip) למודל. על ההגשה לכלול שלושה קבצים:

קובץ המקור (הרחבה של קובץ השלד שניתן) תחת השם AVLTree.java.

קובץ טקסט info.txt המכיל את פרטי המגישים הבאים: תז, שמות ושמות משתמש.

מסמך תיעוד חיצוני, המכיל גם את תוצאות המדידות. את המסמך יש להגיש באחד הפורמטים הבאים: doc, docx או pdf.

שמות קובץ התיעוד וקובץ zip צריכים לכלול את שמות המשתמש האוניברסיטאיים של **שני המגישים** לפי הפורמט AVLTree\_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!