

מבני נתונים - פרויקט מספר 1 - עץ WAVL

דרישות

עליכם לממש עץ WAVL, לפי ההגדרות שניתנו בכיתה. לכל איבר בעץ יש ערך (info) מסוג מחרוזת (String), ומפתח (key) שהוא מספר שלם. כל המפתחות שונים זה מזה. המימוש בשפת ג'אווה וצריך להיות מבוסס על קובץ השלד המופיע באתר הקורס. נגדיר את מספר **פעולות האיזון** להיות מספר $\text{rotation} + \text{promotion} + \text{demotion}$ (double rotation נספר כ-2 פעולות). הפעולות שיש לממש הן:

`empty()` - הפונקציה מחזירה ערך `TRUE` אם ורק אם העץ ריק.

`search(k)` - הפונקציה מחפשת איבר בעל המפתח `k`. אם קיים איבר כזה, היא מחזירה את הערך השמור עבורו, אחרת היא מחזירה `null`.

`insert(k, i)` - הכנסת איבר בעל ערך `i` ומפתח `k` לעץ, אם הוא לא קיים. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ בשלב תיקון העץ על מנת להשלים את הפעולה. אם קיים איבר בעל מפתח `k` בעץ הפונקציה מחזירה -1 ולא מתבצעת הכנסה.

`delete(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.

`min()` - מחזירה את ערכו (info) של האיבר בעל המפתח המינימלי, או `null` אם העץ ריק.

`max()` - מחזירה את ערכו (info) של האיבר בעל המפתח המקסימלי, או `null` אם העץ ריק.

`keysToArray()` - הפונקציה מחזירה מערך ממזין המכיל את כל המפתחות בעץ, או מערך ריק אם העץ ריק.

`infoToArray()` - הפונקציה מחזירה מערך מחרוזות המכיל את כל המחרוזות בעץ, ממוינות על פי סדר המפתחות. כלומר הערך `j` במערך הוא המחרוזת המתאים למפתח שיופיע במיקום `j` במערך הפלט של הפונקציה `keysToArray()`. גם הפונקציה הזאת מחזירה מערך ריק אם העץ ריק.

`size()` - הפונקציה מחזירה את מספר האיברים בעץ.

`select(int i)` - הפונקציה מחזירה את הערך של האיבר עם המפתח ה-`i` בגודלו בעץ, או -1 אם העץ ריק, או שלא קיים צומת עם הדרגה הזאת. (`select` מוגדר במצגת)

`getRoot()` - מחזיר את השורש של העץ (אובייקט `IWAVLNode`)

בנוסף למימוש הפונקציות האלו, יש לממש את מחלקת `WAVLNode` כפי שמתואר בקובץ. ניתן להוסיף מחלקות נוספות, אך כל מחלקה שמייצגת צומת בעץ צריכה לממש את `IWAVLNode` interface.

`IWAVLNode` יש את הפונקציות הבאות:

`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 הימני של הצומת, או `null` אם אין כזה

`getRight` - מחזיר את הבן הימני של הצומת, או `null` אם אין כזה

`isRealNode` - מחזיר כן אם הצומת מייצג צומת אימיתי בעץ (צומת שאינו וירטואלי)

getSubtreeSize – מחזיר את מספר הצמתים האמיתיים בתת עץ של הצומת. יש לממש בסיבוכיות $O(1)$.

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. אין להשתמש באף מימוש ספרייה של מבנה נתונים.

סיבוכיות

יש לתעד את סיבוכיות זמן הריצה (האסימפטוטית) של כל פונקציה, 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

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. לקוד המקור יצורף 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 בעצמיכם! בפרט, כדאי מאוד לממש טסטר, כדי לבדוק את תקינות ונכונות המימוש.

בקובץ שתגישו לא תהיה פונקציית main. אם הצלחתם לקמפל את הפרוייקט לבדו (ללא טסטר), זה סימן שמהו לא נכון במימוש שלכם.

הקוד ייבדק על מחשבי בית הספר על גירסא Java8.

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

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

מדריך לעבודה עם Eclipse (סעיפים 5-9, 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

בסעיף זה נספור את מספר פעולות האיזון שנדרשו בהכנסה ומחיקה של איברים מן העץ.

נגדיר ניסוי כסדרה של פעולות insert עבור $n=i*10,000$ איברים אקראיים שונים ולאחר מכן מחיקת כל האיברים בעץ, כאשר סדר המחיקה הוא מהאיבר הקטן לגדול. כתבו תוכנית (אין צורך להגיש אותה) שתריץ 10 ניסויים עם ערכי i בין 1 ל 10. (כלומר 10,000 איברים, 20,000 איברים וכולי).

בכל ניסוי ספרו את מספר פעולות האיזון שהתבצעו בסה"כ במהלך ההכנסה, ומספר פעולות האיזון שהתבצעו בסה"כ במהלך המחיקה, וחשבו את הממוצע לפי מספר ההכנסות \ מחיקות שבוצעו בעץ ואת המקסימום.

רשמו את התוצאות בטבלה הבאה:

מספר סידורי	מספר פעולות	מספר פעולות האיזון הממוצע לפעולת insert	מספר פעולות האיזון הממוצע לפעולת delete	מספר פעולות האיזון המקסימלי לפעולת insert	מספר פעולות האיזון המקסימלי לפעולת delete
1	10,000				
2	20,000				
...					

פרטו מהן התוצאות שציפיתם לקבל בטבלה על סמך ההסבר התיאורטי של עצי WAVL שנלמד בכיתה, והאם התוצאות שקיבלתם בפועל תואמות את הציפיות. הסבירו את משמעות המדידות שביצעתם.

הגשה

הגשת התרגיל תתבצע באופן אלקטרוני באתר הקורס במודל.

הגשת התרגיל היא בזוגות בלבד!

כל זוג ייבחר נציג **אחד** ויעלה תחת שם המשתמש שלו את קבצי התרגיל למודל. על ההגשה לכלול שני קבצים: קובץ המקור (הרחבה של קובץ השלד שניתן), ומסמך תיעוד חיצוני, המכיל גם את תוצאות המדידות. את המסמך יש להגיש באחד הפורמטים הבאים: pdf, doc, docx, rtf או txt. שמות הקבצים צריכים לכלול את שמות המשתמש האוניברסיטאיים של **שני המגישים** (לדוגמה, WAVLTree_username1_username2.java). 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!