

מבני נתונים – פרויקט מספר 1 – AVL ורשימות

בפרויקט זה נממש את ה-ADT רשימה באמצעות שני מבני נתונים שונים. הראשון הוא המימוש המוכר "מערך מעגלי" שראינו בהרצאות, והשני יהיה מבוסס על עצי חיפוש מסוג AVL (עליהם למדנו כמימוש אפשרי למילון, כאן כאמור הם ישמשו במימוש לרשימה).

המימוש יהיה בשפת ג'אווה וצריך להיות מבוסס על קבצי השלד המופיעים באתר הקורס.

הפעולות אותן נרצה לממש הן הפעולות הבאות (שמהוות חלק מהפעולות המגדירות רשימה). אנו מניחים כי כל איבר ברשימה מכיל שני שדות: מפתח (key) מסוג מספר שלם (int) וערך (info) מסוג מחרוזת (String). נסמן ב- n את מספר האיברים ברשימה ברגע נתון.

- `Retrieve(int i)` - הפונקציה מחזירה את האיבר במיקום i , עבור $0 \leq i < n$. אם i לא בתחום זה יוחזר `null`.
- `insert(int i, int k, String s)` - הכנסת איבר בעל מפתח k וערך s לאינדקס i ברשימה. הפונקציה מחזירה -1 אם $i < 0$ או $i > n$ אחרת הפונקציה מחזירה 0.
- `delete(int i)` - מחיקת איבר באינדקס i ברשימה. הפונקציה מחזירה -1 אם $i < 0$ או $i > n-1$ אחרת הפונקציה מחזירה 0.

1. רשימה מעגלית:

המימוש הראשון יהיה כאמור באמצעות רשימה מעגלית. הרשימה המעגלית תמומש במחלקה `CircularList`. רשימה מעגלית היא ADT רשימה הממומש באמצעות מערך מעגלי כפי שמופיע בשקפים. ודאו שסיבוכיות הפעולות השונות היא כפי שנלמד בכיתה. הרשימה תקבל פרמטר `maxLen` שהוא גודל הרשימה המקסימלי.

דרישות

המימוש של הרשימה המעגלית צריך לתמוך בפעולות `retrieve` ו `delete` כפי שתואר ובפעולה:

- `insert(int i, int k, String s)` - הכנסת איבר בעל מפתח k וערך s לאינדקס i ברשימה. הפונקציה מחזירה -1 אם $i < 0$ או $i > n$ או $n = \text{maxLen}$ אחרת הפונקציה מחזירה 0.

2. מימוש מבוסס AVL:

המימוש השני יהיה כאמור מבוסס על עצי AVL. נקרא למימוש כזה "רשימה עצית".

כעת אנו רוצים שפעולות הרשימה הנ"ל יבוצעו בזמן $O(\log n)$ במקרה הגרוע (המימוש הקודם לא עומד בדרישה זו).

נסו לחשוב כיצד ניתן לבצע זאת באמצעות עצי AVL (רמז: עצי דרגות מסוג AVL). לאחר שניסיתם לחשוב על כך בעצמכם, אתם מחמנים להסתייע בשקפים או בסרטון המצורפים עם התרגיל.

דרישות

תחילה, עליכם לממש עץ AVL, לפי ההגדרות שניתנו בכיתה. העץ ימומש במחלקה `AVLTree` והמימוש שלו ייבדק בפני עצמו. נניח כי כל המפתחות שונים זה מזה, והסדר על צמתי העץ מתייחס כרגיל אך ורק למפתחות.

הפעולות שיש לממש עבור עץ ה-AVL הן:

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

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

insert(int k, String s) - הכנסת איבר בעל ערך s ומפתח k לעץ, אם הוא לא קיים. הפונקציה מחזירה את מספר פעולות האיזון שנדרשו בסה"כ בשלב תיקון העץ על מנת להשלים את הפעולה (גלגולי LR ו-RL נחשבים כ-2 פעולות איזון). אם קיים איבר בעל מפתח k בעץ הפונקציה מחזירה -1 ולא מתבצעת הכנסה.

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.

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

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

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

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

size() - הפונקציה מחזירה את מספר האיברים בעץ.

getRoot() - מחזיר את השורש של העץ (אובייקט AVLNode)

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

AVLNode יש את הפונקציות הבאות:

getKey - מחזיר את המפתח של הצומת.

getValue - מחזיר את info של הצומת.

getLeft - מחזיר את הבן השמאלי של הצומת, או null אם אין כזה

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

getHeight - מחזיר את גובה הצומת. יש לממש בסיבוכיות $O(1)$.

לאחר שמימשתם מחלקה זו תוכלו להתקדם למימוש רשימה עצית. רשימה עצית תמומש במחלקה TreeList. שימו לב שחלק מהפעולות שמימשתם במחלקה AVL יסייעו לכם במימוש רשימה עצית.

בקבצי השלד מופיעים ה header ים של כל הפונקציות. המימוש יבוצע על ידי מילוי קבצי 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 בקבצים המתאימים. אין להוסיף קבצים. 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, כתלות במספר האיברים במבנה אותו סימנו ב- 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 שלכם, קבצי המקור יבדקו 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 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.

בקבצים שתגישו לא תהיה פונקציית main (דבר זה יפגע בטסטר שיבדוק לכם את התרגילים). 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 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>

מידות

בסעיף זה נשווה בין יעילות המימושים של רשימה עצית ומעגלית. נבצע את הניסויים הבאים:

1) נגדיר ניסוי כסדרה של פעולות insert עבור $n=i*10,000$ איברים. בחרו את מיקומי ההכנסה של האיברים כך שידגומו את היתרון של רשימה מעגלית על פני רשימה עצית. כתבו תוכנית (אין צורך להגיש אותה) שתריץ 10 ניסויים עם ערכי i בין 1 ל 10. (כלומר 10,000 איברים, 20,000 איברים וכולי). מדדו את זמני ההכנסה עבור כל אחת מסוגי הרשימות ואת כמות פעולות הגלגול השונות עבור רשימה עצית.

רשמו את התוצאות בטבלה הבאה:

מספר סידורי	מספר פעולות	זמן הכנסה ממוצע עבור רשימה מעגלית	זמן הכנסה ממוצע עבור רשימה עצית	כמות גלגולים ימינה ממוצעת עבור רשימה עצית	כמות גלגולים שמאלה ממוצעת עבור רשימה עצית
1	10,000				
2	20,000				
...					

פרטו מהן התוצאות שציפיתם לקבל בטבלה על סמך ההסבר התיאורטי של רשימה מעגלית ועצית. ועל סמך המיקומים שנבחרו, והאם התוצאות שקיבלתם בפועל תואמות את הציפיות. הסבירו את משמעות המדידות שביצעתם. ציינו מה הם המיקומים שבחרתם ומה הסיבה שהם נבחרו.

(2) נגדיר ניסוי כסדרה של פעולות insert עבור $n=i*10,000$ איברים. בחרו את מיקומי ההכנסה של האיברים כך שידגימו את היתרון של רשימה עצית על פני רשימה מעגלית. כתבו תוכנית (אין צורך להגיש אותה) שתריץ 10 ניסויים עם ערכי i בין 1 ל 10. (כלומר 10,000 איברים, 20,000 איברים וכולי).

מדדו את זמני ההכנסה עבור כל אחת מסוגי הרשימות ואת כמות פעולות הגלגול השונות עבור רשימה עצית.

רשמו את התוצאות בטבלה הבאה:

מספר סידורי	מספר פעולות	זמן הכנסה ממוצע עבור רשימה מעגלית	זמן הכנסה ממוצע עבור רשימה עצית	כמות גלגולים ימינה ממוצעת עבור רשימה עצית	כמות גלגולים שמאלה ממוצעת עבור רשימה עצית
1	10,000				
2	20,000				
...					

פרטו מהן התוצאות שציפיתם לקבל בטבלה על סמך ההסבר התיאורטי של רשימה מעגלית ועצית. ועל סמך המיקומים שנבחרו, והאם התוצאות שקיבלתם בפועל תואמות את הציפיות. הסבירו את משמעות המדידות שביצעתם. ציינו מה הם המיקומים שבחרתם ומה הסיבה שהם נבחרו.

(3) נגדיר ניסוי כסדרה של פעולות insert עבור $n=i*10,000$ איברים. כאשר מיקום ההכנסה מתפלג בצורה אחידה. כלומר אם יש ברשימה N איברים, מיקום ההכנסה של האיבר ה- $N+1$ יוגרל בצורה אחידה בטווח $[0, N]$. בצעו את הניסוי עבור רשימה מעגלית ועבור רשימה עצית.

כתבו תוכנית (אין צורך להגיש אותה) שתריץ 10 ניסויים עם ערכי i בין 1 ל 10. (כלומר 10,000 איברים, 20,000 איברים וכולי).

מדדו את זמני ההכנסה עבור כל אחת מסוגי הרשימות ואת כמות פעולות הגלגול השונות עבור רשימה עצית.

רשמו את התוצאות בטבלה הבאה:

מספר סידורי	מספר פעולות	זמן הכנסה ממוצע עבור רשימה מעגלית	זמן הכנסה ממוצע עבור רשימה עצית	כמות גלגולים ימינה ממוצעת עבור רשימה עצית	כמות גלגולים שמאלה ממוצעת עבור רשימה עצית
1	10,000				
2	20,000				
...					

פרטו מהן התוצאות שציפיתם לקבל בטבלה על סמך ההסבר התיאורטי של רשימה מעגלית ועצית, 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!

הגשה ביחידים תתאפשר רק באישור המתרגלים.

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

1. ארבעת קבצי המקור שניתנו תחת השמות:

- 1. Item.java
- 2. CircularList.java
- 3. TreeList.java
- 4. AVLTree.java

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

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

שמות קובץ התיעוד וקובץ הzip 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!