

מבני נתונים - פרויקט מספר 1 - עץ דרגות

דרישות

עליכם לממש עץ AVL, לפי ההגדרות שניתנו בכיתה. לכל איבר בעץ יש ערך (info) מסוג מחרוזת (String), 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:

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 ממזין 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 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.

split(int x) - הפונקציה מקבלת מפתח x שנמצא בעץ. על הפונקציה להפריד את העץ ל-2 עצי AVL כאשר המפתחות של האחד גדולים מ-x ושל השני קטנים מ-x. יש לממש את הפונקציה על פי המימוש שנלמד בהרצאה בסיבוכיות $O(\log n)$.

join(AVLNode x, AVLtree t) - הפונקציה מקבלת צומת x ועץ t שכל ה-keys שלהם קטנים (או גדולים) מה-keys של העץ הנוכחי. על הפונקציה לאחד את x, t לעץ הנוכחי כפי שמומש בהרצאה. על הפעולה לרוץ בזמן $O(\log n)$. על הפעולה להחזיר את העלות של פעולת ה-join (הפרש גבהי העצים+1).

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. מטעמי נוחות (יקל עליכם לממש גלגולים מכיוון שלכל צומת יהיו 2 בנים), נדרוש שלכל עלה יהיו 2 בנים "וירטואליים", כלומר, צמתים ללא מפתח.

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 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 כן אם הצומת מייצג צומת אמיתי בעץ (צומת שאינו וירטואלי)
 getHeight – מחזיר את גובה הצומת (1- עבור צומת וירטואלי). 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 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.

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 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.

בקובץ 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 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>