

מבני נתונים – פרויקט מס. 2.

ערימה בינומית

מבוא

במטלה זו אתם מתבקשים לממש ערימה בינומית (לא עצלה), כפי שנלמדה בכיתה.

הדרישות

עליכם לממש ערימה בינומית (Binomial Heap), לפי ההגדרות שניתנו בכיתה. תוכלו למצוא את הפרטים במצגת שבאתר הקורס ובCormen.

בתרגיל זה נניח שהאיברים בערימה הם תמיד מספרים שלמים אי-שליליים **ייחודיים** (כלומר אותו מספר לא יכול לחזור יותר מפעם אחת). המימוש צריך להיות מבוסס על קובץ השלד המופיע באתר הקורס. הפעולות שמופיעות בקובץ:

empty(): הפונקציה מחזירה ערך TRUE אם ורק אם הערימה ריקה.

insert(i): הכנסת איבר שערכו i לערימה.

deleteMin(): מחיקת איבר המינימום של הערימה (אין צורך להחזיר אותו)

findMin(): הפונקציה מחזירה את הערך של איבר המינימום בערימה
הפונקציות findMin, deleteMin מניחות שהערמה לא ריקה.

meld(heap2): מיזוג הערימה עם ערימה נוספת heap2.

size(): הפונקציה מחזירה את מספר האיברים בערימה

arrayToHeap(array): בונה ערימה חדשה ממערך של מספרים. כל איברי המערך צריכים להיות בערימה. התוצאה היא ערימה שמכילה את המספרים האלה בלבד (אם היו קודם מספרים אחרים הם ימחקו).

minTreeRank(): הפונקציה מחזירה את דרגת העץ הבינומי הקטן ביותר בערימה.

binaryRep(): הפונקציה מחזירה את הייצוג הבינארי של הערימה. כלומר, היא מחזירה מערך בוליאני, המוגדר מאינדקס 0 עד הדרגה המקסימלית שקיימת בעץ, ועבור כל עץ שקיים בערימה, הערך באינדקס של הדרגה יהיה TRUE.

isValid(): הפונקציה מחזירה ערך TRUE אם ורק אם הערימה היא ערימה בינומית תקינה.
כלומר – כלל הערמה מתקיים, הערימה בנויה מעצים בינומיים תקינים ונעשה שימוש בכל היותר עץ אחד מכל דרגה.

שימו לב – הפונקציה הזאת צריכה לוודא את התקינות **באופן עצמאי** ולא לסמוך על נכונות המימוש של הפונקציות האחרות!

delete(i): מחיקת האיבר שערכו i מהערימה. אם לא קיים בערימה איבר שערכו i פונקציה זו לא משנה את הערמה.

decreaseKey(i, j): אם קיים איבר בערמה שערכו i , אז ערכו יופחת להיות j (ניתן להניח $j \leq i$). אם לא קיים איבר בערמה שערכו i , אז הערמה לא תשתנה.

בפונקציות delete, decreaseKey נשתמש בהנחה שכל האיברים בערימה שונים זה מזה. ניתן להשתמש בטבלת גיבוב כגון HashMap שתמפה לנו בין הערך i של איבר, לבין האיבר עצמו בערימה. כלומר, תחילה נבדוק האם הערך i קיים בתוך ה-HashMap. אם הוא לא קיים, הפונקציה תסתיים. אם הערך i קיים, אזי `HashMap.get(i)` ייתן לנו גישה לאיבר עצמו שבערימה שערכו i . בכיתה ראינו כיצד לממש decreaseKey, delete ממצב זה, שבו יש לנו גישה לאיבר עצמו שבערימה. בניתוח סיבוכיות שבהמשך התרגיל, ניתן להזניח את עלות הקריאות ל HashMap, ולהניח שכל קריאה כזו לוקחת זמן קבוע $O(1)$.

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 שלא מופיעות בשלד), אך אסור לשנות את ההגדרות של הפונקציות לעיל.

סיבוכיות

יש לתעד את סיבוכיות זמן הריצה (במקרה הגרוע) של כל פונקציה, כתלות במספר האיברים בעץ.

עליכם להשיג סיבוכיות זמן ריצה זהה לזאת שנלמדה בכיתה עבור פעולות של מבנה הנתונים (זמני worst case ו-amortized צריכים להיות אלה שנלמדו בכיתה). לגבי פונקציות שלא נלמדו, עליכם להשיג זמן ריצה אסימפטוטי ב-WC טוב ביותר, כך שלא תפגעו בזמן ריצה WC או amortized של אלה שנלמדו.

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 שמשוה 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/1415b/misc/workenv/>

מדריך (סעיפים 9-15):

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

הנחיות לפתיחת חשבון מחשב, למי שמעוניינת לעבוד במעבדת בית הספר:

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

שימוש בג'אוה 8 במעבדות האוניברסיטה:

מדידות

כתבו תוכנית (אין צורך להגיש אותה) שתפעיל את הפעולות שמימשתם, וענו בעזרתה על השאלה הבאה.

על השאלות הבאות יש לענות ללא הסתמכות על הרצת התוכנה:

1. תארו סדרה של m פעולות על המבנה שמימשתם, כך שעבור כל פעולה מבין $m-1$ הפעולות הראשונות, זמן amortized לפעולה קבוע, ואילו עבור הפעולה האחרונה בסדרה זמן הריצה הוא $\Theta(\log m)$.
2. מה זמן הריצה הכולל מבחינה אסימפטוטית של m הפעולות כפונקציה של m ?
3. עבור $m = 1,000, 2,000, 3,000$ מה יהיה הייצוג הבינארי של הערימה בסדרה של m פעולות כפי שתיארתם (i) אחרי הפעולה ה- $(m-1)$ (ii) אחרי הפעולה ה- m ?

על השאלה הבאה יש לענות עם הסתמכות על הרצת התוכנה:

1. הריצו את סדרת הפעולות שתיארתם בסעיף 1. סיפרו את מספר פעולות linking (כלומר, תליה של עץ בינומי אחד על השני) ב- $(m-1)$ הפעולות הראשונות (עבור שלושת ערכי m). הציגו את התוצאות שקיבלתם (מספר ה-linking) והסבירו אותם.
2. הריצו את סדרת הפעולות שתיארתם בסעיף 1 עבור שלושת ערכי m . וודאו שהתוצאות של (i) ו-(ii) הם כפי שציפתם. רשמו במפורש מה החזירה פונקציית binaryRep אחרי פעולה $m-1$ ואחרי פעולה m , והשוו לחישוב התיאורטי 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, BinomialHeap_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!