

מעבדה 10. נושא: ערימת מינימום

תאריך הגשה: 17.01.2023 בשעה 23:00 (בזוגות)

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

במעבדה הנוכחית נממש תור עדיפויות בעזרת ערימה בינארית. את הערימה נממש באמצעות מערך.

תיאור:

ביישומים רבים יש צורך בניהול תור שהוצאת האיברים ממנו תהיה על בסיס של עדיפות. במקרים כאלה תור רגיל לא פותר את הבעיה ביעילות. שימוש ידוע מאוד בתור עדיפויות הוא האלגוריתם של דייקסטרה למציאת מסלול קל ביותר בגרף עם קשתות ממושקלות.

הגדרה 1 (תור עדיפויות):

תור עדיפויות הוא מבנה התומך לפחות בשתי הפעולות הבאות:
insert מכניס איבר למבנה
deleteMin מוצא, מוחק ומחזיר את האיבר המינימלי במבנה.

מימושים אפשריים של תור עדיפויות:

- **רשימה מקושרת:** זמן הכנסה $O(1)$; זמן ליניארי למחיקת המינימום.
 - **רשימה מקושרת ממוינת:** זמן ליניארי להכנסה וזמן קבוע להוצאה.
 - **עץ חיפוש מאוזן** נותן $O(\log N)$ לשתי הפעולות, אך שימוש בעץ חיפוש הוא יותר ממה שצריך.
- נראה מימוש פשוט הנותן הכנסה בזמן לוגריתמי, מחיקה בזמן לוגריתמי, ובנייה בזמן ליניארי. מימוש זה הוא **ערימה (heap)**. הערה: מימוש ע"י ערימה יעיל יותר עבור תור עדיפויות בהיותו ייעודי לבעיה זו, ולכן מאפשר פעולות נוספות בזמן ריצה קצר, למשל, מציאת מינימום בזמן $O(1)$.

הגדרה 2 (ערימה):

ערימה היא עץ בינארי המקיים שני תנאים:
תנאי מבני: העץ צריך להיות כמעט שלם
תנאי סדר: לכל צומת X בעץ, המפתח של ההורה של X קטן או שווה מהמפתח של X

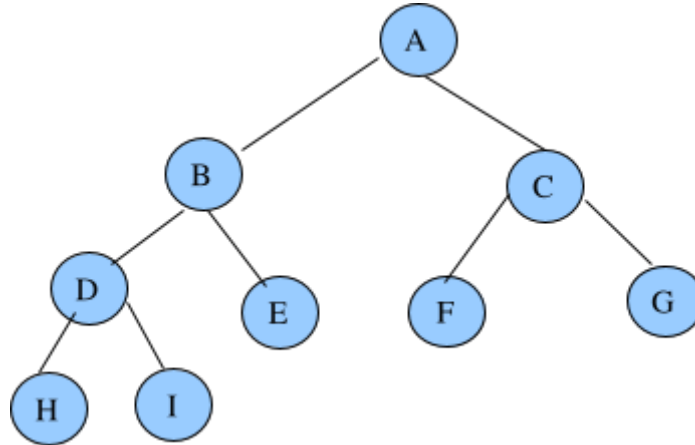
עץ כמעט שלם (complete) הוא עץ מלא פרט אולי לרמה התחתונה, שממולאת משמאל לימין. מספר הצמתים בעץ שלם בגובה h נע בתחום: $2^{h+1} - 1$ — 2^h לכן הגובה הוא $\theta(\log N)$.

תכונת הסדר מבטיחה שמפתח של השורש הוא קטן ביותר, ולכן מציאת המינימום היא בזמן קבוע.

מימוש הערמה:

אפשר לאחסן ערמה במערך, רמה אחרי רמה, החל ממיקום 1. לכל מיקום i במערך, הבן השמאלי הוא במיקום $2i$, והבן הימני במיקום $2i + 1$. מצד שני, מיקום האב לבן הנמצא במיקום i יהיה במקום $\lfloor i/2 \rfloor$.

למשל, העץ הבינארי הבא יוחזק כמערך כדלהלן:



0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I				

הערה: שימו לב, כי המקום ה-0 במערך לא מכיל נתונים של הערימה.

(1) השלימו את המחלקה MinHeap הגנרית

- עם פרמטר שהוא טיפוס המממש את הממשך Comparable
- עם שני בנאים

MinHeap(int), MinHeap(T[])

- והשיטות isFull, isEmpty, insert, getMin, deleteMin, toString.

- כדי ליצור מערך חדש מטיפוס גנרי T בשם data למשל, יש לבצע זאת בצורה הבאה:

`data = (T[]) new Comparable[size];`

שורת קוד זו יוצרת "הערת אזהרה".

כדי שהקומפילר לא יתייחס לאזהרה זו, ניתן לכתוב מעליה:

`@SuppressWarnings("unchecked")`

אם עדיין יש אזהרה בבודק האוטומטי - ניתן להחליף שורה זו בשורה הבאה:

`@SuppressWarnings({"unchecked", "rawtypes"})`

אחד הבנאים יוצר ערימה ריקה בגודל נתון, והשני יוצר ערימה המכילה את כל איברי המערך הנתון.

insert: האיבר החדש ייכנס למקום הבא במערך, ואחר כך יפעפע למעלה כמה שצריך.

delMin: מביאים את האיבר במקום האחרון למקום הראשון (לשורש), ואחר כך יפעפע למטה כמה שצריך.

יש קובץ נוסף ב MOODLE עם הסבר יותר מפורט על הכנסה ומחיקה.
הערה: אם הערימה ריקה, אז deleteMin תחזיר null, ו-getMin תזרוק חריגה.

(2) הוסיפו למחלקה Person, כך שהיא תממש את הממשק Comparable כאשר ההשוואה היא על פי id בלבד.
המשימה היא לגרום למחלקת הבדיקה HeapTest להתקמפל ולעבוד בצורה תקינה.

עבודה נעימה !!!

סדר העבודה ופרטים טכניים

- שלפת הפרויקט DS-Lab10-MinHeap מתוך GITHUB בקישור:

<https://github.com/ykanizo/DSLAb2022-2023Public>

○ אם אין לכם גישה לפרויקט שהורדתם מ GITHUB במעבדות הקודמות יש לבצע שליפה מחדש.

○ אם יש לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה אז בצעו:

- קליק על שם הפרויקט.
- עכבר ימני
- Team-->Pull
- File-->Import->Git->Projects From Git->Existing Local Repository

פורמט קובץ ההגשה ובדיקתו:

פורמט: יש להגיש קובץ ZIP בשם

43_lab10_123456789_987654321.zip

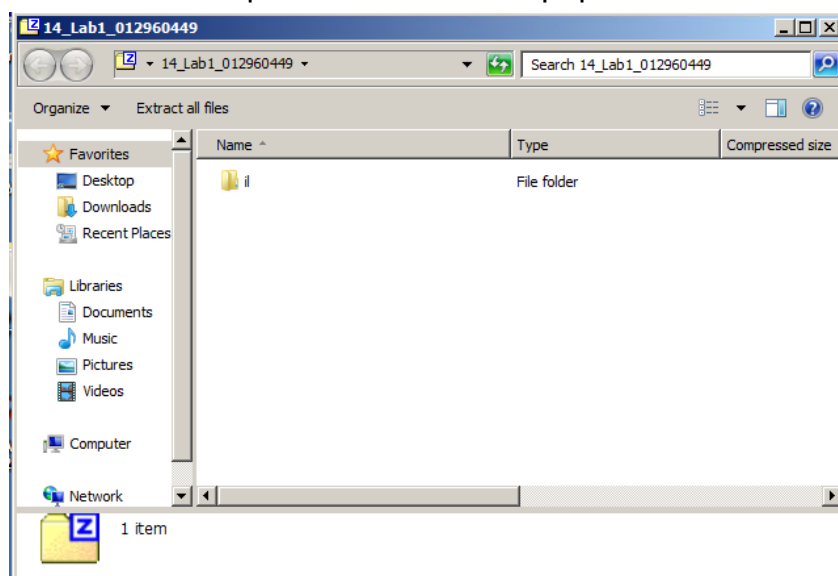
(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

על הקובץ להכיל את כל קבצי ה JAVA שכתבתם כאשר הם נמצאים בתתי תיקיות בתוך התיקייה
il/ac/telhai/ds/

על פי המבנה של הפרוייקט הנתון.

כלומר, השורש של קובץ ההגשה יכיל רק תיקייה בשם il, והוא יכיל את כל קבצי - java על פי התבנית הנתונה בפרוייקט.

להמחשה תמונה של קובץ כזה שנפתח ב - WindowsExplorer



בדיקת קובץ ההגשה: בדקו את הקובץ שיצרתם בתוכנת הבדיקה בקישור:
<https://csweb.telhai.ac.il/>

ראו [סרטון הדגמה](#) של השימוש בתוכנת הבדיקה.

חשוב !!!

בדיקת ההגשות תבוצע ברובה ע"י תוכנית הבדיקה האוטומטית הנ"ל. תוצאת הבדיקה תהייה בעיקרון זהה לתוצאת הבדיקה הנ"ל שאתם אמורים לערוך בעצמכם. כלומר, אם ביצעתם את הבדיקה באתר החוג, לא תקבלו הפתעות בדיעבד. אחרת, ייתכן שתרגיל שעבדתם עליו קשה ייפסל בגלל פורמט הגשה שגוי וכו'. דבר שהיה ניתן לתקנו בקלות אם הייתם מבצעים את הבדיקה. היות ואין הפתעות בדיעבד, לא תינתן אפשרות של תיקונים, הגשות חוזרות וכד'.

הגשה שלא מגיעה לשלב הקומפילציה תקבל ציון 0.

הגשה שלא שמתקמפלת תקבל ציון נמוך מ- 40 לפי סוג הבעיה.

הגשה שמתקמפלת תקבל ציון 40 ומעלה בהתאם לתוצאות הריצה, ותוצאת הבדיקה הידנית של הקוד (חוץ ממקרה של העתקה).

תכנית הבדיקה האוטומטית מכילה תוכנה חכמה המגלה העתקות. מקרים של העתקות יטופלו בחומרה