

מעבדה 11. נושא: גרף

תאריך הגשה: 22/06/2023 בשעה 23:00 (בזוגות)

שימו לב: מעבדה זו תשמש אותנו בשתי המעבדות הבאות, ולכן חובה להגיש מעבדה זו.

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

במעבדה הנוכחית נממש גרף.

תיאור:

גרף הוא מבנה נתונים המשמש לשמירת נתונים וקשרים בין זוגות נתונים. דוגמאות: מפת כבישים בין ערים, מבנה של קשרי חברות ברשת חברתית וכו', מבנה של ויקיפדיה

מבנה של גרף

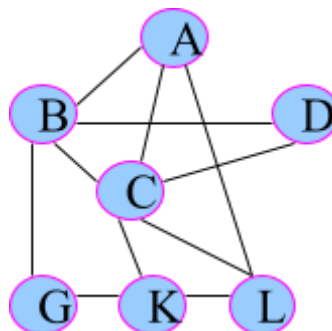
גרף $G = (V, E)$ מורכב מ:

- קבוצה V של קודקודים (Vertices או Nodes)
- קבוצה E של קשתות (Edges)
- כל קשת מחברת בין שני קודקודים.
- קשת e המחברת בין שני קודקודים u ו- v תסומן ב uv .

דוגמא:

$$V = \{A, B, C, D, G, K, L\}$$

$$E = \{AB, AC, AL, BC, BD, BG, CD, CK, CL, GK, KL\}$$



סוגים של גרפים

1. גרף לא מכוון הוא גרף עבורו אין משמעות לסדר הקודקודים בקשתות הגרף, כלומר $uv=vu$.
2. גרף מכוון הוא גרף עבורו יש חשיבות לסדר הקודקודים בקשתות הגרף, כלומר uv שונה מ- vu .
3. גרף הוא ממושקל אם לכל קשת e שלו מוצמד מספר $w(e)$ הנקרא משקל הקשת.

מימושים אפשריים של גרפים:

בעזרת מטריצת סמיכויות - מתאימה במיוחד לגרפים לא דלילים (עם הרבה קשתות).

בעזרת רשימת סמיכויות/פגיעות - מתאימים במיוחד לגרפים דלילים.

במעבדה הנוכחית נממש גרף לא מכוון בעזרת רשימת סמיכויות.

מותר להשתמש ב `java.util` אבל לא ב `java.util.Graph`.

במבנה הזה,

- גרף מורכב מקבוצת צמתים

המלצה: ממשו בעזרת המחלקה `java.util.TreeMap`.

- כל צומת מכיל את רשימת הקשתות הפוגעות בו

המלצה: ממשו בעזרת `java.util.LinkedList`, וטיפוס סטטי `List`.

- וכל קשת מצביעה לצמתים בשתי הקצוות שלה.

יש לכתוב מחלקה גנרית עם שני פרמטרים המתארת גרף (`Graph`). הפרמטר הראשון מציין סוג הצמתים והשני את סוג הקשתות. לדוגמה המחלקה:

```
Graph<Person, Friendship>
```

מתארת אנשים עם יחס חברות ביניהם. המחלקה

```
Graph<City, Double>
```

יכולה לתאר ערים עם מרחקים ביניהם.

שימו לב: בנוסף לשתי המחלקות הנ"ל שמהוות את הפרמטרים של המחלקה, יש צורך במחלקה `Edge`. כדאי ונכון לממש אותה כמחלקה פנימית של `Graph`.

כותרת המחלקה:

```
public class Graph<V extends Comparable<V>, E> implements IGraph<V, E>
```

הגרף ימומש בעזרת השדה:

```
private Map<V, List<Edge>> vertexToIncidence = new TreeMap<V, List<Edge>>();
```

כאשר `Edge` היא מחלקה פנימית ב-`Graph` המתארת קשת ולה שלשה שדות:

```
V u,v; E label
```

גרף יהיה ממושקל, אם המחלקה השניה המזוהה עם הקשתות של הגרף יורשת מ `java.lang.Number` או מממשת את הממשק `Weighted`.

אם `label` יורש מ-`Number` ניתן לקבל את ערכו ע"י ביצוע:

```
((Number) label).doubleValue();
```

תיאור השיטות נמצא בתוך הפרויקט.

אין לשנות את המחלקות שקיבלתם חוץ מהמחלקה Graph.

עבודה נעימה !!!

סדר העבודה ופרטים טכניים

- שליפת הפרויקט DS-Lab11-Graph מתוך GITHUB בקישור:

<https://github.com/michalHorovitz/DSLAb2023Public>

- אם אין לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה יש לבצע שליפה מחדש לפי ההוראות במעבדה הראשונה.
 - 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
- אם אתם עובדים ב VDI, מומלץ לשנות את המיקום המוצע לפרויקט בתיקייה כלשהי בכוון H.

פורמט קובץ ההגשה ובדיקתו:

פורמט: יש להגיש קובץ ZIP בשם

46_lab11_123456789_987654321.zip

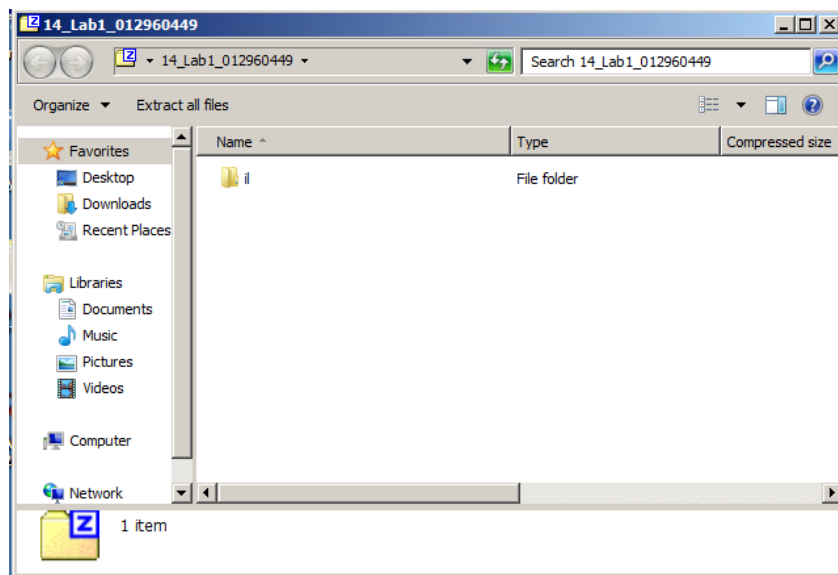
(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

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 מתקמפלת 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).

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