

# מעבדה 12. נושא: מרחקים בגרף ממושקל

תאריך הגשה: 26.01 בשעה 23:00 (בזוגות)

**יש לקרוא היטב לפני תחילת העבודה !**

## מבוא:

במעבדה הנוכחית נוסיף למחלקה גרף שכתבנו במעבדה הקודמת פונקציה שמחזירה את המרחק מצומת נתון, לכל צומת.

## תיאור:

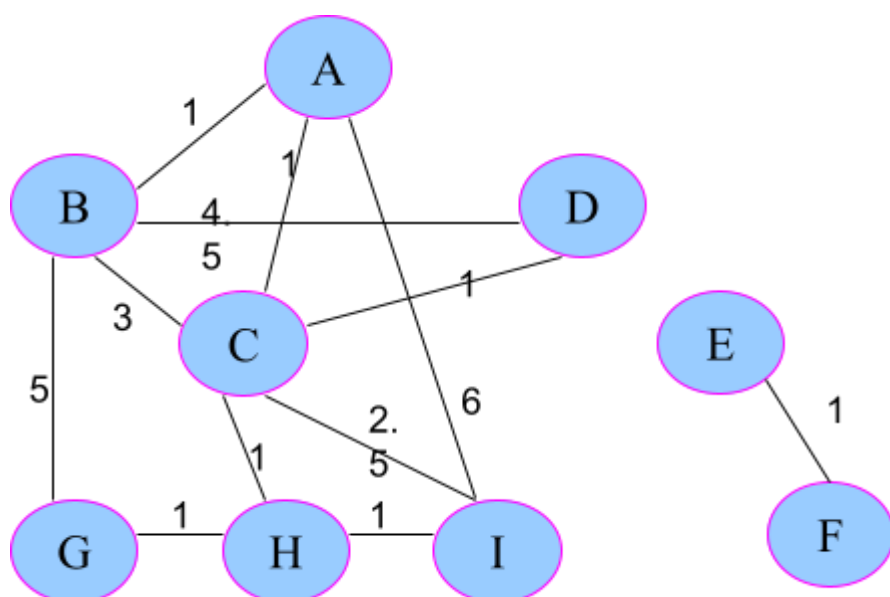
**מסלול בגרף** בין שני צמתים  $u$  ו- $v$  הוא סידרה

$v = v_0 - e_1 - v_1 - e_2 - \dots - e_{k-1} - v_{k-1} - e_k - v_k = v$  כאשר כל  $e_i$  היא קשת בין הצמתים  $v_{i-1}$  ו- $v_i$ . אורך של מסלול בגרף ממושקל הוא סכום משקלות הקשתות שבו:

$$\sum_{i=1}^k w(e_i)$$

**מרחק** בין שני צמתים  $u$  ו- $v$  **בגרף** הוא אורך מסלול קצר ביותר בין שני צמתים אלה.

לדוגמה: המרחק בין  $B$  ל- $C$  בגרף הבא הוא 2 (דרך  $A$ ) למרות שיש ביניהם קשת ישירה. המרחק בין  $C$  ל- $E$  הוא  $\infty$ .



**האלגוריתם של דייקסטרה:** הוא אלגוריתם לחישוב מרחקים מצומת נתון  $s$  בגרף ממושקל כאשר המשקל של כל קשת הוא אי-שלילי. האלגוריתם מחשב את המרחק בין  $s$  לכל צומת בגרף. הוסיפו למחלקה Graph שיטה בשם `distancesFrom` שמקבלת קודקוד בגרף (אובייקט  $v$  מסוג  $V$ ) ומחזירה רשימה של הקודקודים בגרף והמרחקים של  $v$  מכל אחד מהם בפורמט של TreeMap: לכל קודקוד מותאם המרחק שלו מ- $v$ .

אלגוריתם זה משתמש בתור קדימויות של הקודקודים לפי מרחקם מ- $s$ . השמורה (אינווריאנטה) של האלגוריתם היא שבכל פעם שמוציאים קודקוד מהתור, מרחקו הוא סופי ונכון, ולא יתעדכן עוד.

בתחילה התור יכיל רק את  $s$ . בכל פעם שמוציאים קודקוד  $v$  מהתור, נעדכן את המרחקים של השכנים  $u$  של  $v$  אם ניתן להגיע ל  $u$  דרך  $v$  ובכך לקבל מסלול יותר קצר.

#### איתחול:

- המרחק של  $s$  (מעצמו) הוא 0 והמרחק של כל צומת אחר הוא  $\infty$ .
- הכנס את  $s$  לתור קדימויות (לפי מרחק).

האינווריאנטה של האלגוריתם היא "המרחק הקטן ביותר בתור הקדימויות הוא נכון", וכל השאר ייתכן וניתנים לשיפור. היות ואין קשתות שליליות בגרף האינווריאנטה מתקיימת לאחר האתחול, כי המרחק הקטן ביותר בתור הוא 0 (של  $s$ ) וזה כמובן נכון.

#### לולאה:

- הוצא מתור הקדימויות את הצומת  $v$  שמרחקו קטן ביותר
- הכרז על המרחק של  $v$  כסופי (בלתי ניתן לשיפור עתידי)
- עבור כל שכן  $u$  של  $v$  בצע:
- אם ניתן להגיע ל  $u$  דרך  $v$  ובכך לשפר את המרחק של  $u$  אז:
- שפר את המרחק של  $u$
- הוסף את  $u$  לתור הקדימויות שנה את מקומו של  $u$  בתור הקדימויות (ע"י הוצאה אפשרית והכנסה חזרה)

**אין לשנות את המחלקות שקיבלתם חוץ מהמחלקה Graph.**

**עבודה נעימה !!!**

# סדר העבודה ופרטים טכניים

המשיכו את הפרויקט DS-Lab11-Graph שהגשמתם במעבדה הקודמת.

## פורמט קובץ ההגשה ובדיקתו:

**פורמט:** יש להגיש קובץ ZIP בשם

43\_lab12\_123456789\_987654321.zip

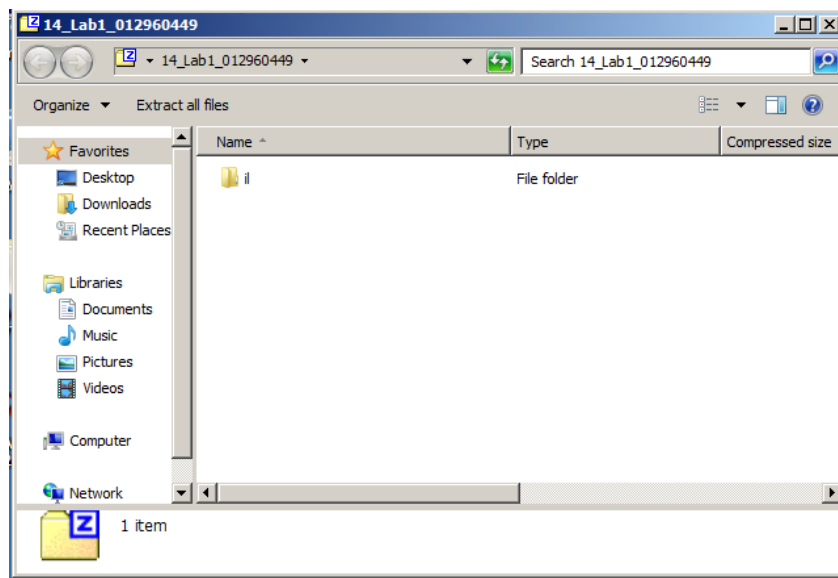
(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

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 יכול רק תיקייה בשם il, והוא יכול 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).

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

