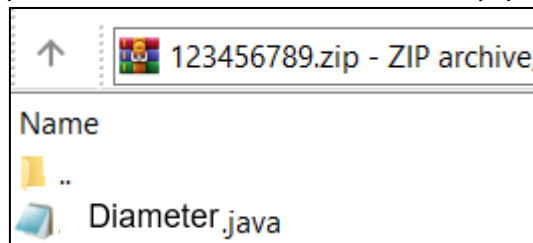


אלגוריתמים 2 סמסטר ב' התשפ"א – תחרות 2

מציאת קוטר של עץ

הנחיות:

- שפת תכנות – JAVA
- הפונקציות שעליכם לכתוב צריכות להיות קודם כל נכונות, ולאחר מכן מהירות ככל האפשר.
- תרגיל זה נעשה ביחידים בלבד.
- מועד אחרון להגשת המטלה: **6.06.21 23:58**.
- תרגיל זה ייבדק בצורה אוטומטית ע"י תוכנית מחשב שתשתמש בשמות הקבצים והפונקציות המוזכרים להלן.
- את התרגיל יש להגיש למקום המתאים במידע האישי שלכם.
- הקובץ לא יהיה מוגדר תחת **package**!
- שם הקובץ: `Diameter.java`
- הקובץ הזה **לבדו** יהיה בתוך קובץ `zip`, ששמו תעודת הזהות שלכם. לדוגמא: `123456789.zip`



- בתחילת הקובץ יש לרשום הערה עם מספר תעודת הזהות שלכם. לדוגמא:

```
// ID: 123456789
public class Diameter {

}
```

- אופן הבדיקה: בשלב הראשון הקוד שלכם נבדק בצורה אוטומטית על אוסף טסטים. רק הסטודנטים שעוברים את כל הטסטים, עוברים לשלב השני: בדיקה מהירות. כאן, מי שמהיר יותר יזכה בניקוד גבוה יותר. כדי להיות מהיר ביותר, תוכלו להשתמש בכל אלגוריתם ומבנה נתונים שלמדנו \ נמצא בספרות \ באינטרנט \ לפתח בעצמכם. אתם מוזמנים לחפש ולבחון את יעילות מבני הנתונים והאלגוריתמים השונים האפשריים.

קובץ שלא יוגש לפי הדרישות הנ"ל לא ייבדק! – אין ויתורים

ניסוח הבעיה:

עליכם למצוא את הקוטר - המרחק הגדול ביותר בעץ. (כמובן בגרף לא מכוון)

הערות:

כדי להקל על המטלה, שמות הקודקודים יתחילו מ 0 (כפי שנהוג ב Java).

קלט:

מטריצת שכנות המייצגת את העץ.

הנחיות לתכנות:

כתבו מחלקה בשם **Diameter** לחישוב המרחק המקסימלי בעץ. מחלקת **Diameter** צריכה להכיל את הפונקציות הבאות:

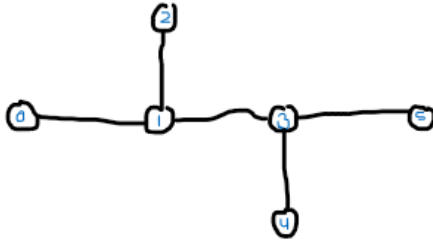
1. בנאי המחלקה **public Diameter(boolean[][] adj_matrix)** מקבל את מטריצת השכנות (אם יש צלע אז מופיע true).
2. פונקציה **public int get_diam()** שמחזירה את הקוטר.
3. אתם יכולים להשתמש בפונקציות \ משתנים נוספים לפי בחירתכם..

שימו לב:

אתם לא חייבים להשתמש במבנה הנתונים שמופיע כקלט (מטריצה – מערך דו ממדי) בחישובים. אתם בהחלט יכולים להמיר אותו למבנה נתונים שיהיה לכם הרבה יותר נח (ומהיר) לעבוד איתו: להשתמש במערכים, רשימות, וקטורים, תורים, מחסניות או כל מבנה נתונים שתמצאו כדי להגיע למהירות מקסימלית. תוכלו גם לבנות מחלקות חדשות משלכם, אך כולן צריכות להופיע בתוך אותו קובץ **Diameter.java. בנוסף, שימו לב שיש אלגוריתמים שונים בעלי סיבוכיות שונה שיכולים לעזור לכם להצליח במטלה (מומלץ מאוד לנסות כמה אלגוריתמים שתמצאו ולאחר מכן לשלוח את הטוב ביותר).**

ניתן להשתמש בדוגמה הבאה לבדיקת נכונות התוכנית.

Graph



Input

False	True	False	False	False	False
True	False	True	True	False	False
False	True	False	False	False	False
False	True	False	False	True	True
False	False	False	True	False	False
False	False	False	True	False	False

Output

get_diam() = 3

עבודה מהנה!