

קורס מונחה עצמים – מטלה 3: גרפים ב Python

במטלה זו נממש מבנה נתונים של גרף ממושקל ומכוון ב Python, המימוש כולל מחלקה של גרף וכן מחלקה של אלגוריתמים על גרפים. הרעיון המרכזי הוא להיעזר במימוש הקודם שלכם (של מטלה 2) ו"לתרגם" אותו ל Python, ולאחר מכן להשוות את ביצועי הפתרון שלכם למימוש הקודם שלכם ב java. **יודגש אסור להשתמש בספריות קיימות לגרפים (בפרט NetworkX), במימוש הפתרון שלכם למטלה זו.**

כל המידע ומחלקות העזר למטלה זו נצאים בקישור זה: [Ex3](#)

חלק ראשון:

מימוש המחלקה של גרף: השתמשו בהגדרה של [GraphInterface](#) כדי לממש את המחלקה DiGraph. לאחר שהשלמתם את המחלקה, בדקו את עצמכם בעזרת הפונקציה check0() בקובץ [main.py](#). לאחר שבדקתם שכל השמות נכונים, ממשו מחלקת בדיקה בשם TestDiGraph – שתבצע בדיקה יחידה (UnitTesting) של כל שיטה במחלקה.

חלק שני:

ממשו את המחלקה GraphAlgo שיורשת מהמחלקה (האבסטרקטית) [GraphAlgoInterface](#). בדקו את עצמכם ע"י הרצה של הפונקציות check1(), check2() בקובץ [main.py](#). לאחר שבדקת שכל השמות נכונים, ממשו מחלקת בדיקה בשם TestGraphAlgo שתבצע בדיקה יחידה (UnitTesting) של כל שיטה במחלקה. טיפ:

- כדאי לממש את האובייקט Node
- על מנת להאיץ את shortest_path מומלץ לממש OrderdSet ו-Path
- לא להסתבך עם plot_graph, צריך להיות מסודר, אבל אין צורך לעשות גרפיקה יפה במיוחד אלא תצוגה בסיסית שתאפשר debug של המימוש שלכם.

חלק שלישי:

בחלק זה עליכם לבצע השוואה של הקוד שכתבתם למול המטלה הקודמת (Ex2) שלכם (ב java), ההשוואה צריכה לכלול את המרכיבים הבאים:

3.1 בדיקת נכונות – על אותם קבצים, כדי לאפשר בדיקה השוואתית של זמן ריצה על אותם תרחישים – בדגש על אלגוריתמים.

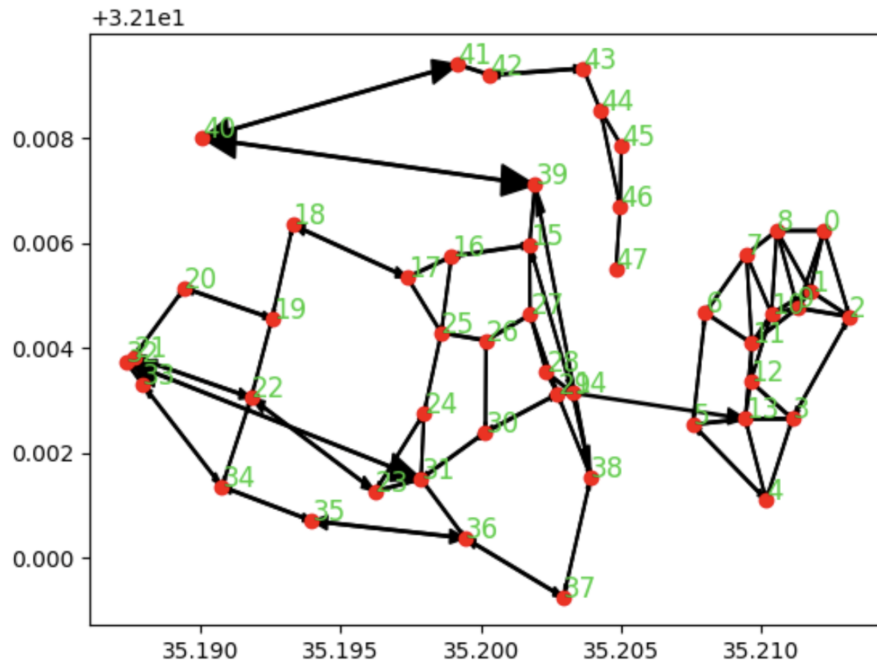
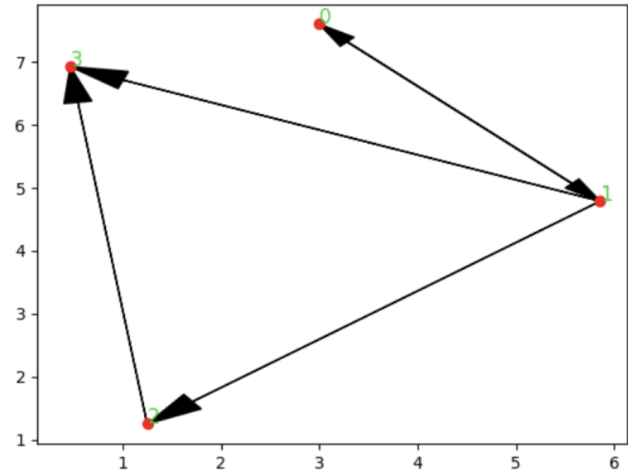
• יש להשוות זמני ריצה, נא לציין מפרט מחשב (מערכת הפעלה, דגם מעבד, זיכרון, וכו' צבע פחות רלוונטי)

• לצרף גרפים של התוצאות, ניתן להשתמש בmatplotlib או באקסל (לא לצייר על דף ולצלם)

• יש להשוות את הפונקציות הבאות:

- Load & save
- shortest_path
- centerPoint
- TSP

איור 1: צילום מסך של התוכנית [main.py](#) (פונקציה check1) שמציגה גרף עם 4 קודקודים (ללא מיקומים קבועים, אלא כאילו שנבחרו באקראי).



איור 2: צילום מסך של התוכנית Ex3_main (פונקציה check2) שמציגה גרף עם 47 קודקודים (הקובץ A5 מהמטלה הקודמת שכולל מיקומים קבועים – ובהתאם צרכים להיות מוצגים בהתאם).

הנחייה כללית:

- מטלה זו מוגדרת בעיקר ע"י מספר "ממשקים" שמגדירים את ה api הנדרש מהמחלקות שאתם נדרשים לממש, לנוחיותכם מימשנו עבורכם קובץ בדיקת שמות בשם Ex3_main, אנא עשו בו שימוש לבדיקת השמות של המטלות שלכם.
- את הדוח ההשוואתי יש לסכם כמסמך שייכתב כדף wiki בפרויקט ה github שלכם.
- חובה לעשות שימוש ב UnitTesting, לכל מחלקה לוגית שאתם כותבים.
- אין לשנות את הקבצים (המחלקות האבסטרקטיות) שקיבלתם.
- המטלה מיועדת לזוגות – חשוב מאוד ללמוד לעבוד בזוגות ע"ג אותו repository.
- הנחיות הגשה: את המטלה יש להגיש כפרויקט github, בקישור [הבא](#) עד ה 28.12.2021
- זוהי מטלה רביעית (ולפני אחרונה) בקורס, הקפידו לתעד היטב את הפתרון שלכם, כתבו Readme מפורט ודפי wiki שכוללים הסברים על מבנה הקוד, אופן מימוש האלגוריתמים, כיצד להוריד ולהריץ וכמובן השוואת ביצועים למול מטלה 2 שלכם.

בהצלחה!