# מטלת פייתון – numstack

אפשר לענות על חלק מהשאלות ולקבל ניקוד חלקי.

## 1. פתרון משוואות: השוואת זמן ריצה [2 נק']

ראינו שתי ספריות המאפשרות לנו לפתור מערכות של משוואות ליניאריות: numpy ו cvxpy. מטרת השאלה היא לבדוק איזו ספריה מהירה יותר.

א. כתבו קוד היוצר אוטומטית מערכות של משוואות ליניאריות בגדלים שונים, עם מקדמים אקראיים.

ב. פתרו כל אחת מהמערכות שיצרתם בעזרת numpy ובעזרת cvxpy.

ג. שרטטו גרפים של זמן הריצה כפונקציה של גודל המערכת.

איזו ספריה מהירה יותר, ובכמה?

## 2. בדיקת איכות של אלגוריתמי קירוב [2 נק']

בספריה networkx יש לא רק אלגוריתמים מדוייקים, אלא גם אלגוריתמי קירוב מהירים.

מצאו אלגוריתם אחד כזה (חפשו approximations בתיעוד של הספריה).

א. בדקו מהו יחס הקירוב התיאורטי של האלגוריתם שמצאתם (היעזרו בתיעוד, או במאמרים שהתיעוד מפנה אליהם).

ב. הפעילו את האלגוריתם על גרף אקראי במודל G[n,p] עם ערכים שונים של n, p. לכל הפעלה, חשבו את יחס הקירוב של האלגוריתם על הגרף. שימו לב: כדי לחשב יחס קירוב, צריך לדעת את הפתרון המדוייק. יש שתי דרכים לעשות זאת:

* להשתמש באלגוריתם מדוייק כלשהו, גם אם הוא מאד לא יעיל, למשל חיפוש שלם.
* לשנות מעט את הגרף האקראי כך שתוכלו לדעת בוודאות את הפתרון המדוייק.

ג. שרטטו את יחס הקירוב שמצאתם כפונקציה של n, עבור ערכים שונים של p, ב-subplots שונים. עד כמה יחס הקירוב טוב יותר מהיחס התיאורטי?

שאלה 3: משחק תיכנות [1 נק']

בחרו שאלה אחת חדשה (שעדיין לא פתרתם) מהרשימה הבאה: <https://www.codingame.com/training>

בדרגת קושי **קשה מאד – Very Hard**. פתרו אותה בפייתון.

יש להגיש קישור לפתרון שלכם באתר codingame + צילום מסך.

תוספת [+1 נק']: חפשו שאלה, שכדי לפתור אותה צריך להשתמש באלגוריתם משמעותי כלשהו של networkx או scipy או cvxpy.