פרקטיקום

נתיחס לעץ בינארי Tree כלרשימה [root, left-tree, right-tree]. עם אינרפיס הבא:

```
def Nil():
    return []

def Tree(root, L, R):
    return [root, L, R]

def root(t):
    return t[0]

def L(t):
    return t[1]

def R(t):
    return t[2]
```

נגדיר את המוסגים הבאים:
תת-עץ הוא עץ המורכב מקדקוד כלשהו וכל הצאצאים שלו.
עומק של איבר בעץ הוא מספר האבות הקדמונים שלו.
רמה היא רשימה של כל הקדקודים הבעלים אותו עומק.
משקל של תת-מבנה (ענף, תת-עץ, רמה, ...) הוא סכום של הערכים בקדקודים שלו.
כמות של קדקודים העץ t נסמן כ- $N(t)$.
עץ בינרי נקרא **כימעט חיובי** אם כמות הקדקודים החיוביים בו גדולה מהכמות הקדקודים אי-שליליים.
עץ בינרי **מאוזן** מוגדר באופן הבא:

- עץ ריק או עץ בעל קדקוד אחד הוא מאוזן.
- עץ t מעוזן אם $L(t)$ ו- $R(t)$ מאוזנים ו- $|h(L(t)) - h(R(t))| \leqslant 1$.

כל המבנים בשאלות הבאות מכילים את המספרים השלמים.

שאלה 1 (60 נק'). כתבו פונקציה $\mathrm{F}(t)$ שמקבלת את עץ t ומחזירה את תת-עץ הכבד ביתר בין תתי-עצים המאוזנים וכימעט חיוביים של t.
דרישה לסיבוכיות זמן: $O(N(t))$

שאלה 2 (30 נק'). כתבו פונקציה $\mathrm{G}(t)$ שמקבלת את עץ t ומחזירה את המשקל של הרמה הכבדה ביתר ב-t.
דרישה לסיבוכיות הממוצעת זמן: $O(N(t)\log(N(t)))$

שאלה 3 (30 נק'). כתבו פונקציה $\mathrm{H}(t)$ שמקבלת מטריצה A (כ- numpy מערך) ומחזירה את מטריצה D כך ש-
$$D[i,j] = \rho(A[i:], A[j:])$$

כאשר

$$\rho(x,y) = \frac{\sum_{i=0}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n}(x_i - \bar{x})^2 \sum_{i=0}^{n}(y_i - \bar{y})^2}}, \quad \bar{a} = \frac{1}{n}\sum_{i=0}^{n} a_i$$

דרישה: אין להשתמש בלולאות גנרטורים או map