

מטלה – חלוקת שכר-דירה

יוסי טויטו – 314732637

שאלה 4: השמה ממקסמת סכום ערכים - סיבוכיות זמן ריצה

א. מהי סיבוכיות זמן הריצה של האלגוריתם הטוב ביותר המוכר כיום למציאת השמת חדרים לדיירים הממקסמת את סכום הערכים? (לפי ויקיפדיה).

לפי ויקיפדיה, האלגוריתם הטוב ביותר כיום למציאת השמת חדרים לדיירים הממקסמת את סכום הערכים הוא **האלגוריתם ההונגרי**.

One of the first polynomial-time algorithms for balanced assignment was the Hungarian algorithm. It is a *global* algorithm – it is based on improving a matching along augmenting paths (alternating paths between unmatched vertices). Its run-time complexity, when using *Fibonacci heaps*, is $O(mn + n^2 \log n)$,^[2] where m is a number of edges. This is currently the fastest run-time of a *strongly polynomial* algorithm for this problem. If all weights are integers, then the run-time can be improved to $O(mn + n^2 \log \log n)$, but the resulting

מקור: https://en.wikipedia.org/wiki/Assignment_problem

זמן הריצה של האלגוריתם על פי ההרצאה ועל פי ערך הויקיפדיה שלו הוא $O(n^3)$.

The **Hungarian method** is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later *primal-dual methods*. It was developed and published in 1955 by **Harold Kuhn**, who gave the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians: **Dénes König** and **Jenő Egerváry**.^{[1][2]}

James Munkres reviewed the algorithm in 1957 and observed that it is (strongly) polynomial.^[3] Since then the algorithm has been known also as the **Kuhn–Munkres algorithm** or **Munkres assignment algorithm**. The time complexity of the original algorithm was $O(n^4)$, however **Edmonds** and **Karp**, and independently **Tomizawa** noticed that it can be modified to achieve an $O(n^3)$ running time.^{[4][5][now?]}

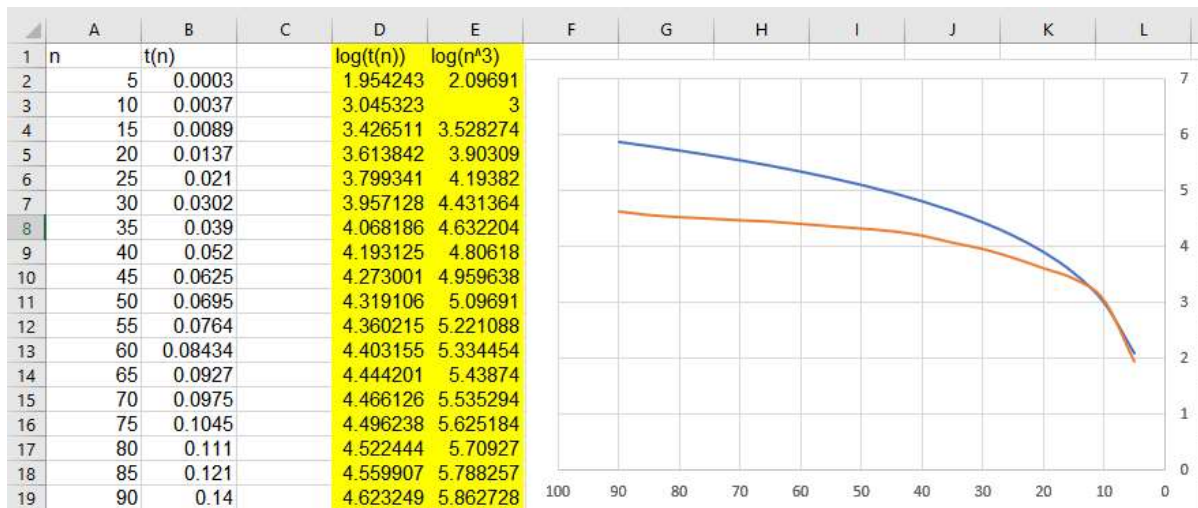
מקור: https://en.wikipedia.org/wiki/Hungarian_algorithm

* ב [רשות]. בדקו ע"י ניסוי, מהי (בקירוב) סיבוכיות זמן הריצה של האלגוריתם `max_weight_matching` הממומש בספריה `networkx` של פייתון? (דרך אפשרית לבדוק היא להריץ את האלגוריתם על גרפים בגדלים שונים, לשרטט גרף של זמן הריצה כפונקציה של גודל הגרף, ולנסות להתאים פונקציה לגרף).

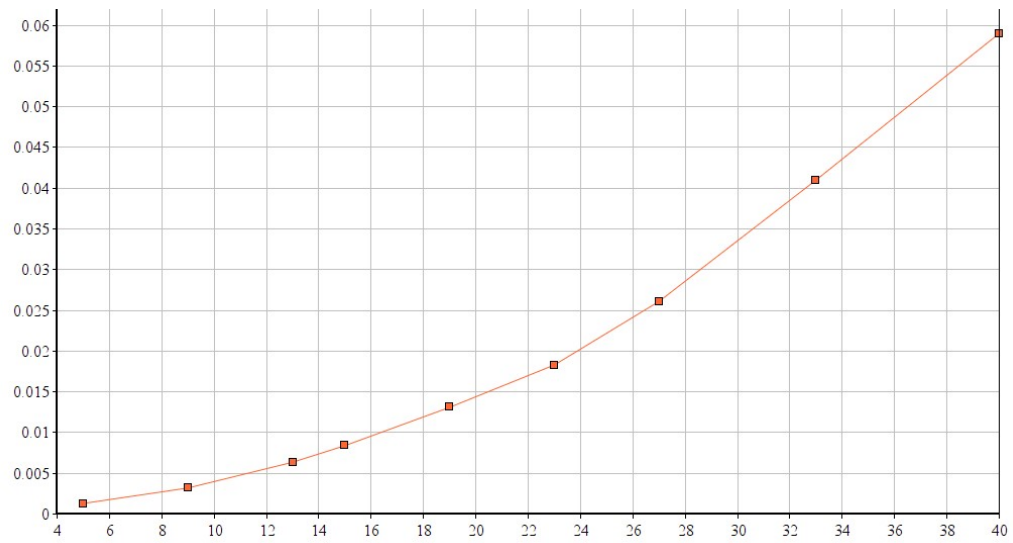
כתבתי קוד בפייתון באמצעות הספריה `networkx`, מדדתי את זמני הריצה של `max_weight_matching` והכנסתי את הערכים למייצר גרפים, ולאקסל. תמונות מצורפות בהמשך ובעמוד הבא.

קישור לקוד שכתבתי: <https://github.com/YosefTwito/EconomicAlgos-Week4.git>

ניתן לראות בתמונת האקסל שלכתחילה זמן הריצה דומה מאוד עבור קלט קטן, וככל שגדל הקלט שמים לב שזמן ריצה מעט קטן יותר מ $O(n^3)$.



תוצאות הקוד שלי במייצר גרפים:



ציור של גרף באמצעות Desmos – גרף עבור x^3 :

