

## פתרון תרגיל 2 - דו"ח \ מגישות: נועם פדות 315097113 ורותם גידליה 211914668

**הסבר הרצה:** כדי להריץ את האלגוריתם צריך להיות מותקן במחשב python3, בנוסף, לביקשתכם יצרנו קובץ exe הנקרא ex2 ומכאן שורת ההרצה:

### ex2 PARAMETER

כאשר להרצת סעיף א PARAMETER יהיה regular להרצת סעיף ב בגרסת דארווין PARAMETER יהיה darwin ולהרצה בגרסת לאמרק PARAMETER יהיה lamarck במידה ותרצו לצפות בגרפים צריך להוריד את ההערות המסמנות קוד לגרף ע"י GRAPHS CODE ולוודא כי החבילה matplotlib מותקנת ובנוסף כדי לצפות במספר hits צריך לעדכן את המפתח האמיתי לפענוח בלולאה הראשית (שורה) שימו לב כי הקבצים של enc.txt וקבצי תדירויות: Letter2\_Freq.txt, dict.txt צריכים להיות באותו תיקייה עם קובץ הקוד.

### דוגמאות הרצה עם הפרמטרים:

```
C:\Users\noamp\Desktop\יב\ex2 darwin
Gen NO. 0 / 300
Gen NO. 1 / 300
Gen NO. 2 / 300
```

```
C:\Users\noamp\Desktop\יב\ex2 regular
Gen NO. 0 / 300
Gen NO. 1 / 300
```

```
C:\Users\noamp\Desktop\יב\ex2 lamarck
Gen NO. 0 / 300
Gen NO. 1 / 300
```

כעת, נתאר את האלגוריתם הגנטי הכללי:

האלגוריתם רץ 300 איטרציות כאשר כל איטרציה עוברים על סט הפתרונות בגודל 100 ולפי פונקציית fitness לוקחים את החצי הטוב של הפתרונות. לאחר מכן עוברים על החצי הטוב ועבור 2 פתרונות רנדומלים מבצעים בהסתברות פעולת crossover ופעולות מוטציה ומוסיפים את הפתרונות החדשים לסט הפתרונות החדש. ממשיכים כך עד שסט הפתרונות מתמלא שוב למקסימום-500.

נסביר איך מימשנו את המרכיבים המרכזיים-

1. **איך יוצגו הפתרונות:** פתרון מיוצג כרשימה באורך 26 של תווים מהאלף בית באנגלית כאשר התו במקום ה-*i* מייצג את ההחלפה של התו ה-*i* באלף בית (0 עבור A, 1 עבור B וכו') לדוגמא: נציג את 5 התווים הראשונים של מפתח (פתרון) `pwdtr` מפתח זה אומר שבטקסט המוצפן כל אות *a* תתחלף באות *p* וכל אות *b* תתחלף באות *w* וכו'.

2. **מהי פונקציית ההערכה:** הפונקציה שמעריכה את טיב הפתרון היא פונקציית fitness שנותנת ציון לפתרון, היא מקבלת את הפענוח של הטקסט לפי מפתח ("פתרון") ועוברת על כל המילים בטקסט המופענח ועבור כל מילה אם היא מופיעה במילון מילים נתון מוסיפה 0.1 לציון ובנוסף עבור כל מילה עוברת על זוגות אותיות ובודקת לפי הקובץ את התדירות של זוג האותיות ומוסיפה את התדירות לציון.

3. **איך ביצעתם את פעולת ה cross-over בין פתרונות:** ממרחב הפתרונות נבחרים 2 פתרונות אקראיים  $p_1$ ,  $p_2$  ובהסתברות של 0.3 מתבצעת פעולת crossover. בפעולת crossover נבחר אינדקס אקראי *i* שאומר לחתוך את  $p_1$  עד אינדקס *i* ולחבר את  $p_2$  מאינדקס *i* וכך גם את הרישא עד *i* של  $p_2$  לחבר עם החצי השני של  $p_1$  מאינדקס *i* והלאה. בעקבות החיתוך וחיבור מחדש ייתכן

ויהיו כפילויות של אותיות ולכן בכל פעולת crossover גם מתבצע תיקון שמחליף עבור אות שחוזרת על עצמה באות שלא נמצאת.

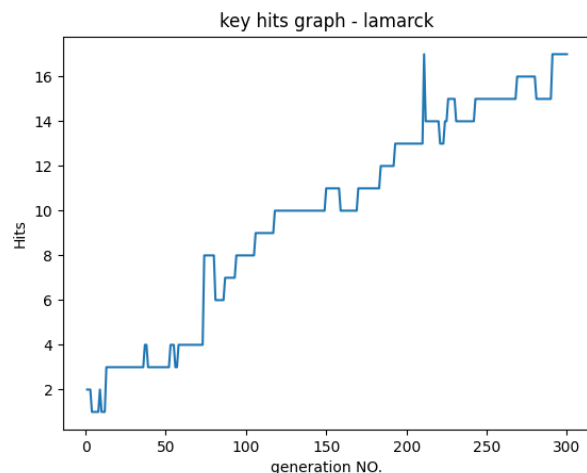
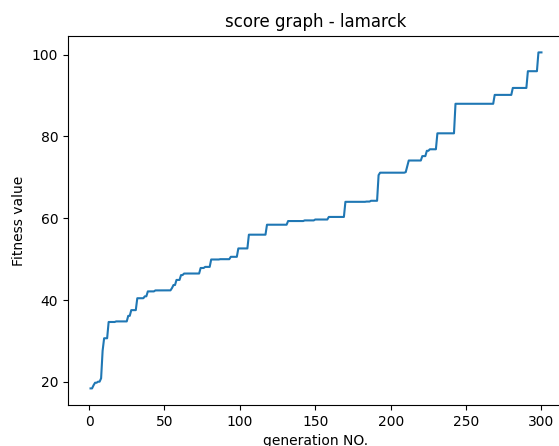
4. **כיצד מומשו מוטציות:** מוטציה התבצעה בהסתברות של 0.1 כלומר עבור פתרון עברנו על כל התווים של הפתרון ועבור כל תו בהסתברות 0.1 נחליף עם תו אחר רנדומלי בפתרון.

5. **האם/איך התייחסתם לבעיית ההתכנסות המוקדמת:** אחד הדברים למזעור בעיית ההתכנסות המוקדמת היא ביצוע מוטציות באופן הסתברותי על הפתרונות, בכך שאנחנו מייצרים פתרונות חדשים אנחנו משנים רנדומלית את מרחב הפתרונות ובכך ממזערים את האופציה להתכנסות מוקדמת, אך אכן ייתכן שנתכנס מוקדם כיוון שביצוע המוטציות ופעולת ה crossover הם הסתברותיים.

6. **איך החלטתם מתי לעצור את הריצה:** אנו נעצור את הריצה לאחר סיום מספר איטרציות קבוע מראש.

השוואת ריצות בגרסאות השונות (בכותרת הגרף רשום הגרסה):

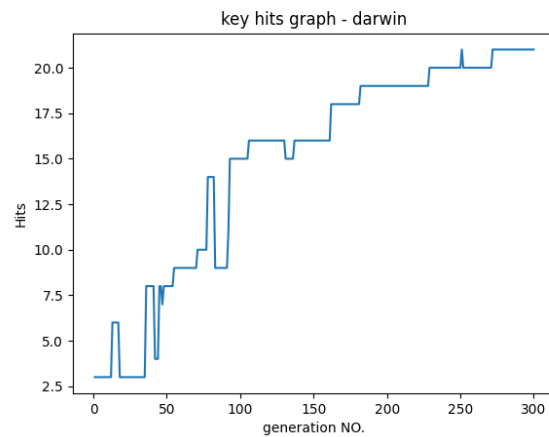
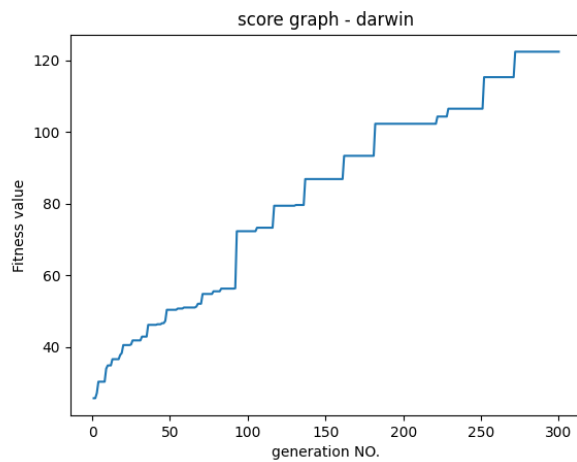
מוצגים 2 גרפים לכל גרסה, גרף אחד המייצג את מספר הפגיעות הנכונות בניחוש המפתח בכל דור והגרף השני מייצג את הציון של פונקציית ה fitness עבור המפתח הכי טוב.



//Lamarck – console log

Hits NO. : 17

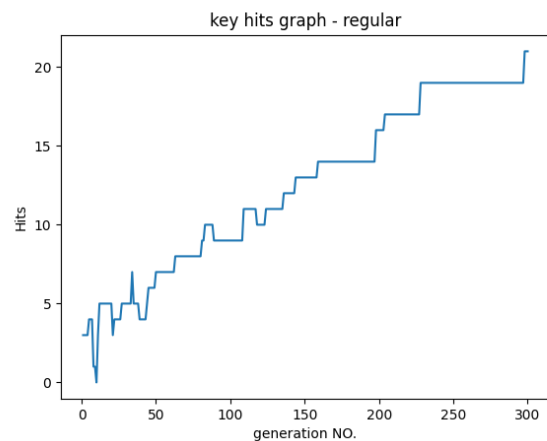
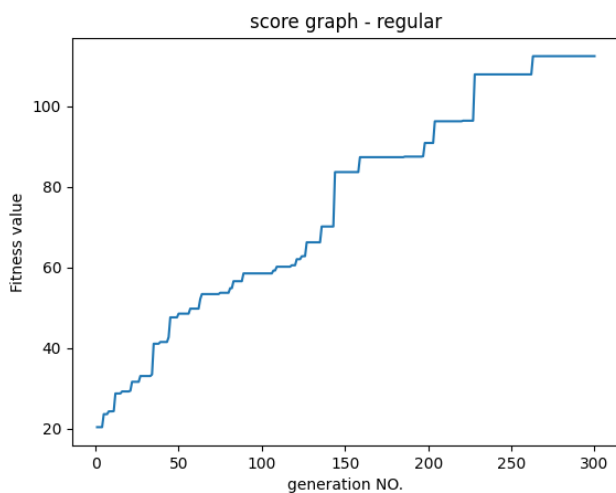
Execution time of find the key: 147.07847809791565 seconds  
Fitness Counter: 33183



// Darwin – console log

Hits NO. : 21

Execution time of find the key: 147.4260401725769 seconds  
Fitness Counter: 33013



// regular - console log

Hits NO. : 21

Execution time of find the key: 135.9890341758728 seconds  
Fitness Counter: 30099

ניתן לראות כי ככל שעולים במספר הדורות הציון fitness עבור המפתח הכי טוב עולה וכמובן בהתאם מספר התווים הנכונים במפתח עולה.

כעת נראה השוואה של התוצאות של האלגו' עם שינויי פרמטרים:

▪ פרמטר גודל אוכלוסייה- מרחב הפתרונות

	regular	lamarck	darwin
POPULATION_SIZE = 100	Hits NO. : 21	Hits NO. : 17	Hits NO. : 21
	Execution time of find the key: 135.9890341758728 seconds	Execution time of find the key: 147.07847809791565 seconds	Execution time of find the key: 147.4260401725769 seconds
	Fitness Counter: 30099	Fitness Counter: 33183	Fitness Counter: 33013
POPULATION_SIZE = 50	Hits NO. : 8	Hits NO. : 19	Hits NO. : 17
	Execution time of find the key: 72.44970798492432 seconds	Execution time of find the key: 84.12899994850159 seconds	Execution time of find the key: 81.07936692237854 seconds
	Fitness Counter: 15349	Fitness Counter: 18387	Fitness Counter: 18331

ניתן לראות כי הקטנת מרחב הפתרונות שיפרה את זמן ריצת האלגוריתם אך רמת הדיוק בפתרון ירדה. צפינו לתוצאות אלו כיוון שככל שמקטינים את המרחב הפתרונות לאלגוריתם ייקח פחות זמן כי עוברים על פחות פתרונות אבל הדיוק ירד. עם זאת ניתן לראות כי הגרסאות של darwin Lamarck מספקות תוצאות טובות יותר וזה קורה כיוון שבגרסאות אלו יש אופטימיזציות לוקאליות שהוספו על האלגוריתם הרגיל.

▪ פרמטר תדירות ביצוע מוטציות

	regular	lamarck	darwin
MUTATION_RATE = 0.1	Hits NO. : 21	Hits NO. : 17	Hits NO. : 21
	Execution time of find the key:	Execution time of find the key:	Execution time of find the key:

	135.9890341758728 seconds	147.07847809791565 seconds	147.4260401725769 seconds
	Fitness Counter: 30099	Fitness Counter: 33183	Fitness Counter: 33013
MUTATION RATE = 0.5	Hits NO. : 8	Hits NO. : 5	Hits NO. : 6
	Execution time of find the key: 139.6461112499237 seconds	Execution time of find the key: 161.62256002426147 seconds	Execution time of find the key: 158.29821634292603 seconds
	Fitness Counter: 30099	Fitness Counter: 33131	Fitness Counter: 33139

ביצוע של יותר מדי מוטציות בפתרונות גורם לשינוי הפתרונות בצורה תדירה יותר כל דור ולכן מספר הפגיעות קטן בכל גרסה, ביצוע יותר מדי שינויים בפתרונות מונע מאפשרות להתכנס לפתרון הנכון ביותר כיוון שאנחנו מאבדים פתרונות טובים בדרך ע"י כך שאנחנו משנים אותם.