דו"ח פרויקט – אלגוריתם אבולוציוני לפתרון קובייה הונגרית

שמות המגישים:

רן עזרא, ת.ז 302779657

נדב חנם, ת.ז 313260648

1. סקירה כללית – קובייה הונגרית

הקובייה ההונגרית המוכרת היא פאזל מכני תלת – מימדי, כאשר כל אחת משמת פאותיה מחולקת לתשעה אריחים כך שכל אריח בצבע שונה מתוך סט של 6 צבעים אפשריים, וכל פאה יכולה לנוע עם או נגד כיוון השעון.

מטרת השחקן היא לבצע סדרת פעולות אשר תוביל למצב שבו כל פאה של הקובייה מורכבת מצבע אחד בלבד.

הקובייה ההונגרית נקראת על שמו של הפסל והפרופ' לאדריכלות ההונגרי "ארנה רוביק" אשר המציא אותה בשנת 1974.

לא פעם עלה הרעיון שהמשיכה הבינלאומית והצלחות היצוא של הקובייה ההונגרית הפכו לאחד מהתורמים המרכזיים להשתקמות והליברליזציה של הכלכלה ההונגרית בין 1985 -1981 אשר בסופו של דבר הוליכה את המעבר מקומוניזם לקפיטליזם.

מדי שנה, מתקיימת תחרות לפתרון קובייה הונגרית בזמן מינימאלי, השיא הנוכחי הרשמי של פתרון קובייה הונגרית הוא **3.47** שניות, הושג על ידי "יושנג דו" בן ה – 21 מסין ב – 24.11.2018.

Yusheng Du

2. **<u>תיאור הבעיה:</u>**

קובייה הונגרית נתונה בגודל 3*3*6 יכולה להימצא ב - 43,252,003,274,489,856,000 תצורות אפשריות.

האתגר העיקרי בפתרון הקובייה הוא העובדה שהאריחים השונים בקובייה תלויים זה בזה, ולא ניתן להזיז שום אריח בנפרד, וגם לא להחליף בין זוג אחד בלבד של אריחי-קצה או זוג של אריחי פינה, גם לא לאחר ביצוע רצף כלשהו של מהלכים כלשהם.

ניתן להוכיח מתמטית כי כל תצורה נתונה של הקובייה ניתנת לסידור כקובייה מושלמת בלא יותר מ – 20 מהלכים, ולכל הפחות ב - 19 מהלכים.

לאור העובדה שקיימות בקובייה 6 פאות בדיוק, וכל פאה ניתנת לסיבוב ב –90° עם או נגד כיוון השעון, קיימים 12 מהלכים אפשריים לביצוע בכל שלב.

בהינתן n מהלכים, קיימים 12^n מהלכים אפשריים.

3. **<u>הסבר על האלגוריתם:</u>**

א. כללי:

האלגוריתם מגריל סידור חוקי כלשהו של קובייה הונגרית.

ע"מ לוודא שהסידור אכן חוקי, מתבצע מס' קבוע של מהלכים חוקיים על קובייה מושלמת, שנבחרים באופן רנדומלי.

בהינתן הקובייה, האלגוריתם מחפש באופן אבולוציוני את רצף המהלכים <u>הקצר</u> ביותר שיביא לכמות מקסימאלית של אריחים הנמצאים על הפאה הנכונה.

בסיום הריצה, האלגוריתם מחזיר כפלט סימולציה של ביצוע רצף המהלכים הטוב ביותר שהושג על הקובייה שהוגרלה.

כל מהלך אפשרי מיוצג ע"י אינדקס בין 0-11, ע"פ המפתח הבא:

אינדקס	מהלך
0	הזזת הפאה העליונה עם כיוון השעון
1	הזזת הפאה העליונה נגד כיוון השעון
2	הזזת הפאה התחתונה עם כיוון השעון
3	הזזת הפאה התחתונה נגד כיוון השעון
4	הזזת הפאה הקרובה עם כיוון השעון
5	הזזת הפאה הקרובה נגד כיוון השעון
6	הזזת הפאה הרחוקה עם כיוון השעון
7	הזזת הפאה הרחוקה נגד כיוון השעון
8	הזזת הפאה השמאלית עם כיוון השעון
9	הזזת הפאה השמאלית נגד כיוון השעון
10	הזזת הפאה הימנית עם כיוון השעון
11	הזזת הפאה הימנית נגד כיוון השעון

ב. התנהגות אבולוציונית:

- <u>ייצוג הפרט:</u>

פרט באוכלוסייה מיוצג ע"י מערך של מספרים בין 0 – 11 בגודל n,
כאשר n הוא מספר הצעדים המקסימאלי המותר.
כל ערך מייצג ביצוע פעולה כלשהי על הקובייה.
נדגיש כי הפרט מציע n מהלכים, אולם ניתן לבחור מתוכם m מהלכים
בלבד, המוגדרים ע"י רישא של המערך בגודל $0 \le m \le n$.

לדוג, עבור $n = 8$, $m = 5$, הפרט

0	3	5	4	8	4	4	2

מייצג את רצף המהלכים

0	3	5	4	8

- <u>חישוב ה – Fitness:</u>

בהינתן פרט (מערך בגודל קבוע מראש), הקובייה המקורית שהוגרלה
מועתקת. כל אחד מהמהלכים ע"פ סדר הופעתם בפרט מבוצע על העותק
של הקובייה ולאחר כל מהלך, מתבצעת ספירה של כמות האריחים
בקובייה שנמצאים בפאה הנכונה. עבור כל פאה מושלמת, מתווסף
לסכום שהתקבל 2.
ערך ה – Fitness הוא הערך המקסימאלי שהתקבל במהלך הספירה.
ערכו יהיה לכל היותר 66 (54 אריחים שמסודרים במקומם, בנוסף ל – 6
פאות מושלמות).

- <u>Selection:</u>

בחירת "טורניר" – מוגרל מס' קבוע של פרטים מתוך האוכלוסייה
הנוכחית, ונבחר מתוכם הפרט עם ה – Fitness הגבוה ביותר.

- <u>Elitism:</u>

בסיום כל דור, הפרט עם ה – Fitness המקסימאלי מבין האוכלוסייה
הנתונה, ממשיך לדור הבא.

- <u>Crossover:</u>

בחנו שיטות Crossover שונות, בסופו של דבר בחרנו
ב 1-point Crossover סטנדרטי.

- <u>Mutation:</u>

בהיתן אינדקס j המייצג את הרישא האידיאלית של הפרט
ועבור פרמטר p כלשהו,
ההסתברות להחלפה תהיה p עבור $i \le j$ ו- 0.1p, אחרת.

I	0	1	2	3	4	5	6	7
Val	6	3	5	4	8	4	4	2
Mutation Prob	0.1p	0.1p	0.1p	0.1p	0.1p	p	p	p

4. **<u>סקירת התוכנה:</u>**

א. המבנה הכללי של האלגוריתם הוא אבולוציוני. ניתן לבצע מס׳ הרצות ברצף של האלגוריתם על קובייה זהה או להגריל קובייה חדשה בכל הרצה, וכן להשוות בסיום כל הרצה את התוצאות שהושגו לחיפוש אקראי בגודל Generations * Population Size.

ב. <u>The Class Cube:</u>

קובייה מתוארת ע״י אובייקט מסוג Cube, המכיל 6 מערכים בגודל 9.
המערכים מתוארים ע״י מיקומם ביחס למי שמתבונן בקובייה ע״י המילים (close ,far ,left ,right ,bottom ,up).
ערכם של התאים במערכים הוא מספר בין 0 ל – 5 המייצג צבע כלשהו, תא מס׳ 4 בכל מערך לא משתנה והוא קובע את ערכו של האריח האמצעי בקובייה.

Init()	מאפשרת יצירת קובייה חדשה, הקובייה שמתקבלת היא קובייה מושלמת.
Randomize_cube()	מבצעת n מהלכים על הקובייה. ניתן להגדיר בחירה אקראית של מהלכים או לחילופין סט קבוע של מהלכים (רלוונטי כאשר רוצים להשוות את ביצועי האלגוריתם על אותה קובייה בדיוק).
Print_cube()	מבצעת הדפסה של הקובייה.
Calculate_cube()	מקבלת פרט (מערך המייצג רצף מהלכים), ומבצעת את הרישא האידיאלית שלו על הקובייה.
Count()	הפונקציה סופרת את מספר האריחים שנמצאים במקום שבו הם צריכים להיות על קובייה נתונה.
Weigthed_Count()	הפונקציה סופרת את מספר האריחים שנמצאים במקום שבו הם צריכים להיות על קובייה נתונה, על כל פאה מושלמת מוסיפה 2 לספירה.

ג. <u>Cube Functions:</u>

מאגר של פונקציות המאפשרות לבצע פעולות שונות על קובייה נתונה.
מתחלקות ל – 3 סוגים:

4 * ()__to close	שינוי נקודת המבט על הקובייה. הערך שמופיע במקום __ יהפוך להיות הפאה שממוקמת ממול למי שמסתכל על הקובייה.
12 * ()1/2__	הערך שמופיע במקום __ מייצג פאה, עבור הערך 1 – הפאה תסובב עם כיוון השעון ועבור הערך 2 – נגד כיוון השעון.
13*_ _s	סט של כ – 5 מהלכים רצופים על הקובייה. המהלכים הם מהלכים שנלקחו מתהליך הפתרון החישובי של קובייה הונגרית.

5. <u>סקירת התוצאות שהושגו</u>

א. <u>בחירת פרמטרים:</u>

<u>מס' דורות:</u>

בכל אחת מההרצות שביצענו, הבחנו כי הערך 150 מהווה חסם עליון למס' הדור שבו מושג שיפור.

לכן, הגבלנו את מס' הדורות ל – **150**.

<u>גודל הפרט:</u>

ידוע כי ניתן לפתור קובייה בכל מצב לאחר לכל היותר 20 מהלכים.

נתון זה היווה בעבורנו השראה לבחירת גודלו של כל פרט, ובהרצות הראשוניות, גודל הפרט הוגדר להיות סביב המספר 20.

לאור תוצאות נמוכות, ולאור העובדה שהחלטנו להגדיר מחדש את המושג פרט כמערך בגודל של n תאים,

מתוכו מתבצעים m פעולות בלבד כך ש $0 \le m \le n$,

פרטים ארוכים יותר אפשרו לנו יותר גמישות, ובתהליך האבולוציה נבחנים פתרונות בכלל האורכים האפשריים.

אל מול הגדרת הפרט באופן זה, גודלו של פרט הוגדר להיות **100**.

<u>ההסתברות למוטציה:</u>

כל פרט מייצג רצף פעולות, לכן שינוי של כל תא בו משפיע על כלל הפעולות שמבוצעות אחריו. שינוי בתאים הראשונים במערך יכול למעשה לשנות לחלוטין את טיבו של הפרט (לטוב ולרע).

לכן, העדפנו לבצע מוטציות בהסתברות נמוכה יחסית.

בסופו של דבר, לאחר הגדרה מחדש של Fitness ופרט באוכלוסייה, ההסתברות לביצוע מוטציה הוגדרה כ – **0.01** על תאים ברישא האידיאלית, ו – **0.1** בתאים שאחריה.

בסופו של דבר, בחרנו לעבוד עם הפרמטרים הבאים:

Population Size	100
Generations	150
Individual Size	100
Tournament Size	10
Mutation Probability	0.1

ב. אבני דרך

בחירת סט הפונקציות שמהן מורכב פרט:

Set1:

בתחילה, הגדרנו את סט הפונקציות הניתנות לביצוע ע"י פרט מסוים באופן חלקי: סיבוב הפאה העליונה של הקובייה ב – 90° עם/נגד כיוון השעון, וסיבוב הקובייה כולה כך שניתן להסתכל על כל אחת מהפאות בתור ה"פאה העליונה".

תיאורטית, אך ורק בעזרת פונקציות אלו ניתן לבצע כל פעולה מתוך 12 הפעולות החוקיות הניתנות לביצוע. רצינו לבחון האם האלגוריתם יוכל לבצע תהליך למידה אבולוציוני ו"ללמוד" כיצד לבצע מהלכים מורכבים יותר, רק בעזרת מהלכי בסיס אלו.

Set2:

הגדרנו מחדש את סט הפונקציות כ – 12 הפעולות החוקיות הניתנות לביצוע (הזזה של כל אחת מהפאות עם או נגד כיוון השעון), וקיבלנו קפיצה דרמטית בביצועי האלגוריתם.

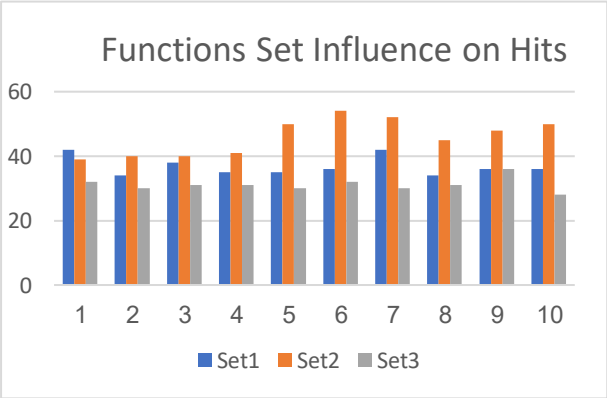
Set3:

בתהליך הפתרון המתמטי של הקובייה, עבור כל מצב נתון שבו הקובייה נמצאת, מוגדר סט של כ – 5 פעולות בממוצע אותן יש לבצע, ע"מ להתקדם לעבר פתרון הקובייה.

ניסינו להחליף את 12 המהלכים המקוריים בסטים של פעולות אלו, מתוך ידיעה ששימוש בהם בחוקיות מסוימת יביא בהכרח לפתרון הקובייה.

ע"מ לבחור Set פונקציות אידיאלי, ביצענו על קובייה זהה 10 הרצות של כל Set, **המדד שנבדק הוא מס' האריחים של הקובייה שהיו במקומם, מתוך 54 אריחים.** קיבלנו את התוצאות הבאות:

	Set3	Set2	Set1
1	32	39	42
2	30	40	34
3	31	40	38
4	31	41	35
5	30	50	35
6	32	54	36
7	30	52	42
8	31	45	34
9	36	48	36
10	28	50	36
Average	**36.8**	**45.9**	**31.1**
S.D	**2.07**	**5.60**	**2.97**



מכאן, הסקנו כי Set2 טוב יותר מ Set1,Set3 ובחרנו להשתמש בו.

אופן חישוב ה – Fitness:

תחילה, הגדרנו Fitness באופן אינטואיטיבי – מס' האבנים שממוקמות במקום הנכון בסיום הפעלת כלל המהלכים שמוגדרים ע"י הפרט.

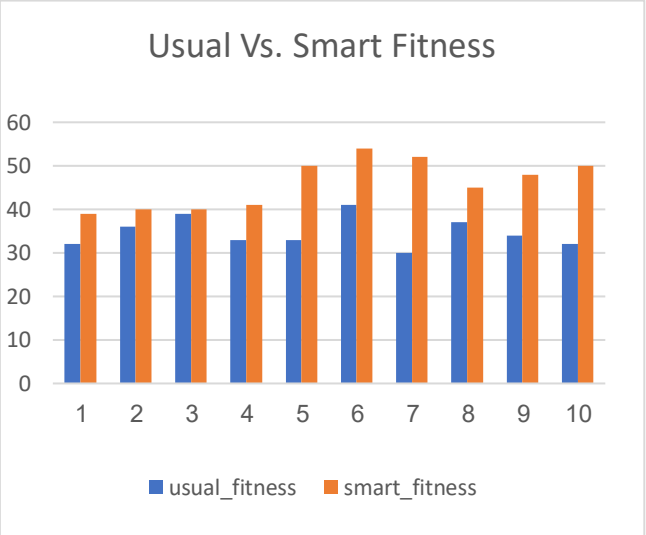
לאחר מכן, ביצענו את האופטימיזציות הבאות:

- סביר להניח שבעת פתרון בעיית הקובייה ע"י גורם אנושי, הוא לא יגדיר לעצמו מראש שהוא מתכוון לבצע בדיוק 20 או 50 מהלכים, ושאם לאחר ביצוע מס' כלשהו של מהלכים הוא יגיע לפתרון, הוא יעצור. הנחה זו היוותה עבורנו השראה לחישוב ה – Fitness בצורה יותר מתוחכמת: לאחר כל צעד שבוצע ע"י הפרט, נבחן את מצבה של הקובייה, והמצב הטוב ביותר שהושג בדרך נשמר כערך ה – Fitness. כמובן שמושקע כוח חישוב רב יותר (עבור כל פרט, מצב הקובייה נבחן לאחר כל מהלך ולא רק בסיומה).
- בנוסף, רצינו "לעודד" את האלגוריתם לגרום לפאה אחת לפחות להיות מושלמת, לאחר שהעמקנו בפתרון המתמטי והבנו שזהו צעד הכרחי בדרך לפתרון הקובייה.

לכן, בחרנו להוסיף לערך ה – Fitness 2 עבור כל פאה מושלמת.

ע"מ לבחור שיטת Fitness אידיאלית, ביצענו על קובייה זהה 10 הרצות של כל שיטה, **המדד שנבדק הוא מס' האריחים של הקובייה שהיו במקומם, מתוך 54 אריחים.** התוצאות שהתקבלו:

	Smart Fitness	Usual fitness
1	39	32
2	40	36
3	40	39
4	41	33
5	50	33
6	**54**	41
7	52	30
8	45	37
9	48	34
10	50	32
Average	**45.9**	**34.7**
S.D	**5.60**	**3.46**



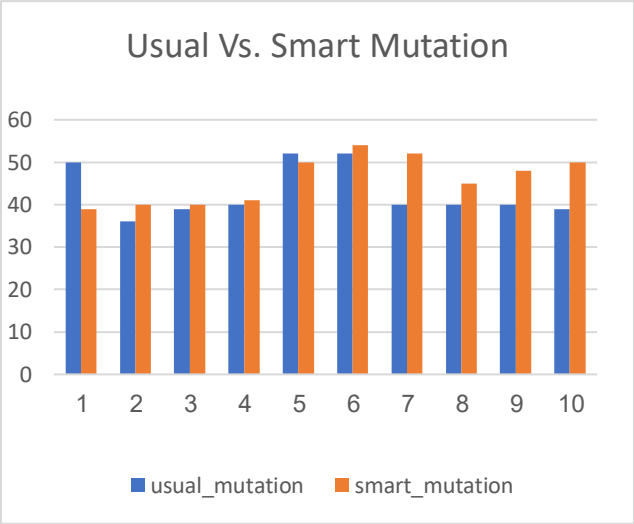
ניתן לראות ששינוי ה – Fitness השפיע באופן משמעותי על הישגי האלגוריתם.

<u>הגדרת מוטציה:</u>

ע"מ להשיג שיפור נוסף בביצועים, החלטנו להתחשב באינדקס של הרישא מתחילת המערך המייצג את הפרט שהשיגה את התוצאה הטובה ביותר בעת ביצוע מוטציה על פרט, כמפורט בהרחבה בחלק הקודם. **המדד שנבדק הוא מס' האריחים של הקובייה שהיו במקומם, מתוך 54 אריחים.**

קיבלנו את התוצאות הבאות:

	Smart mutation	Usual mutation
1	39	50
2	40	36
3	40	39
4	41	40
5	50	52
6	**54**	52
7	52	40
8	45	40
9	48	40
10	50	39
Average	**45.9**	**42.8**
S.D	**5.60**	**6.03**



שינוי שיטת המוטציה לא השפיע בצורה דרמטית על הישגי האלגוריתם.

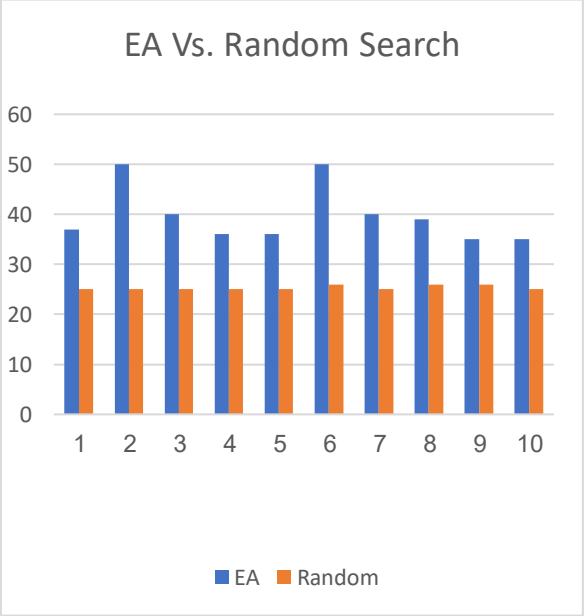
ג. **<u>השוואה אל מול אלגוריתם חיפוש אקראי:</u>**

לטובת ההשוואה, הוגרלו 10 קוביות באופן אקראי והשוונו את תוצאותיו של כל אלגוריתם על כל אחת מהקוביות.

המדד שנבדק הוא מס' האריחים של הקובייה שהיו במקומם, מתוך 54 אריחים.

EA	
Population Size	100
Generations	150
Individual Size	100
Tournament Size	10
Mutation Probability	0.1
Random Search	
Individual Size	100
Number of individuals	Generations * Population Size = 15,000

	Random	EA
1	25	37
2	25	50
3	25	40
4	25	36
5	25	36
6	26	50
7	25	40
8	26	39
9	26	35
10	25	35
Average	**25.3**	**39.8**
S.D	**0.483045892**	**5.692099788**



ניתן לראות בצורה מובהקת שבמהלך כלל הריצות, האלגוריתם האבולוציוני הגיע להישגים טובים יותר.

6. תובנות ולקחים:

התאמת האלגוריתם האבולוציוני לבעיה -

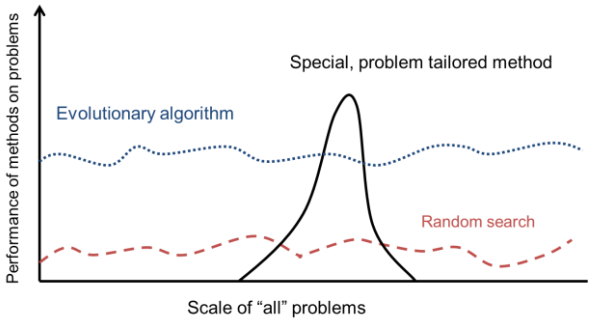
בתחילת התהליך, ניתן לומר שהשתמשנו באלגוריתם אבולוציוני גנרי שעניינו הגרלת רצף אקראי של מספרים וביצוע המהלכים אותם הם מייצגים על קובייה הונגרית.

התוצאות שהושגו היו קרובות לתוצאות שהושגו ע"י הפעלת חיפוש אקראי.

ככל שביצענו יותר התאמות (שיטות המוטציה, חישוב ה – Fitness, הגדרת הפונקציות), השגנו שיפור אל מול התוצאות הקודמות.

במילים אחרות, ככל שביצענו התאמות שנגזרות מאופי הבעיה לאלגוריתם, האפקטיביות של האלגוריתם עלתה.

כלומר, התובנה שראינו במהלך ההרצאות, לפיה טיבו של אלגוריתם אבולוציוני מושפע מסוג הבעיה אותה הוא מנסה לפתור מקבלת משנה תוקף.



השפעתה של פונקציית ה Fitness -

במטרה להשיג תוצאות טובות יותר, ניסינו למטב כל אחד מהמרכיבים האבולוציוניים. ברוב המקרים, השגנו שיפורים מינוריים ובמקרים מסוימים, לא השגנו כלל שיפור. ההשפעה הדרמטית ביותר ו"קפיצת המדרגה" של האלגוריתם קרתה כאשר ניסינו לבצע אופטימיזציה של פונקציית ה – Fitness.

נראה שמבין כלל המרכיבים של אלגוריתם אבולוציוני, המרכיב העיקרי שמשפיע על טיבו של האלגוריתם הוא פונקציית ה – Fitness.

לראייתנו, היא זו שמכווינה את אופן ההתפתחות של האלגוריתם האבולוציוני, וככל שהיא תוגדר בצורה שנותנת את המשקל הנכון לכל פרמטר, כך האלגוריתם יהיה אפקטיבי יותר.

בסופו של דבר, הבנו כי כנראה ש"מסתתרת" אי – שם פונקציית Fitness מושלמת והיא זו שמפרידה בין האלגוריתם שלנו לבין אלגוריתם שיגיע כמעט בכל הרצה לפתרון מושלם, ולא בטוח שנוכל בעזרת תהליך חשיבה אנושי בלבד להגיע אליה (בדיוק הרציונאל שמאחורי שיטת SAFE).

7. ביבליוגרפיה

- "אתר הקובייה ההונגרית", אורי צציק – https://rubiks.glove.co.il/
- Prof. moshe sipper – tiny_ga.py: https://github.com/moshesipper/tiny_ga/blob/master/tiny_ga.py
- ויקיפדיה – "קובייה הונגרית": https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%91%D7%99%D7%99%D7%94_%D7%94%D7%95%D7%A0%D7%92%D7%A8%D7%99%D7%AA