**מבוא לבינה מלאכותית**

**מעבדה 4**

**סטודנט 1 : שם: אניס סרוג'י**

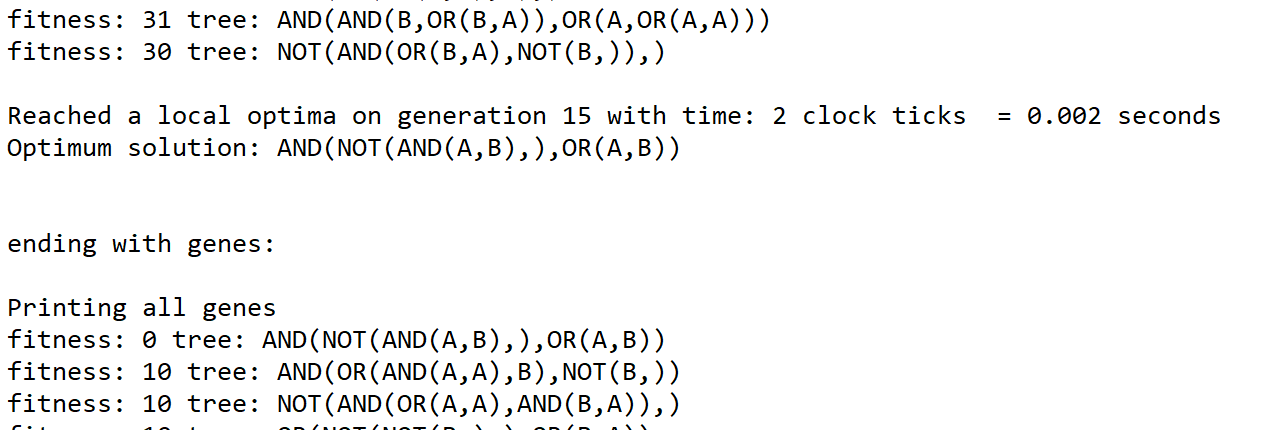
**ת.ז : 206052201**

**סטודנט2 : שם: עומר חטיב**

**ת.ז : 211706445**

**חלק א**

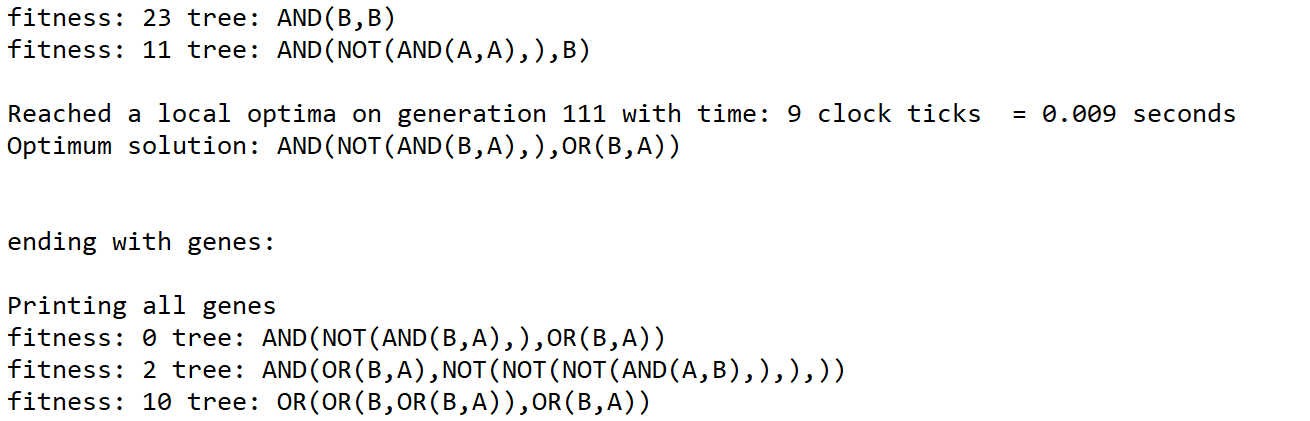
1. הקוד כתוב ומצורף עם העבודה
2. שוב הקוד כתוב , הינה כמה טסטים שעשינו על סט קטן של גנים:



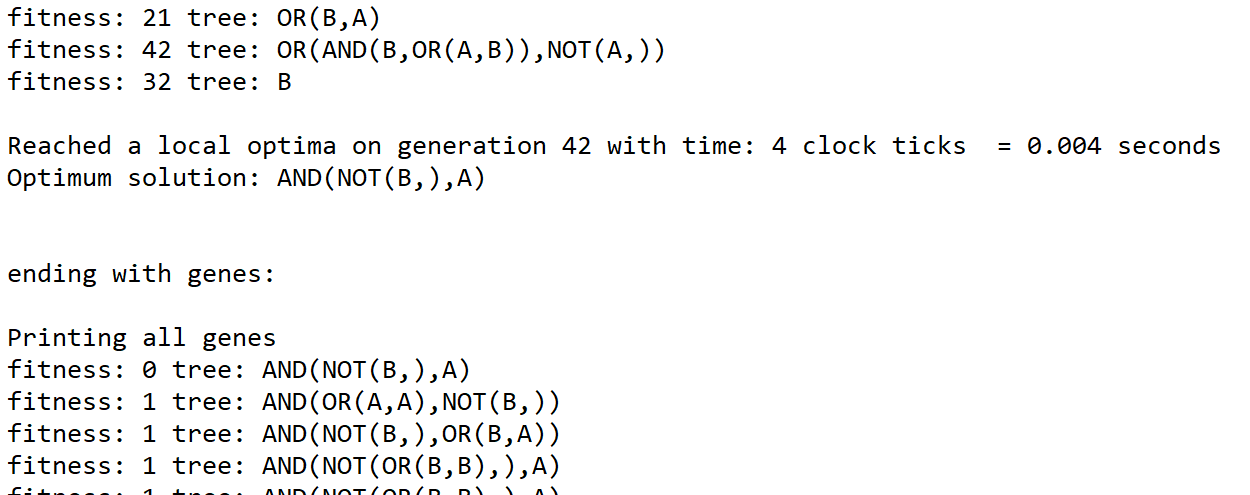
תמונה 2

A picture containing bird

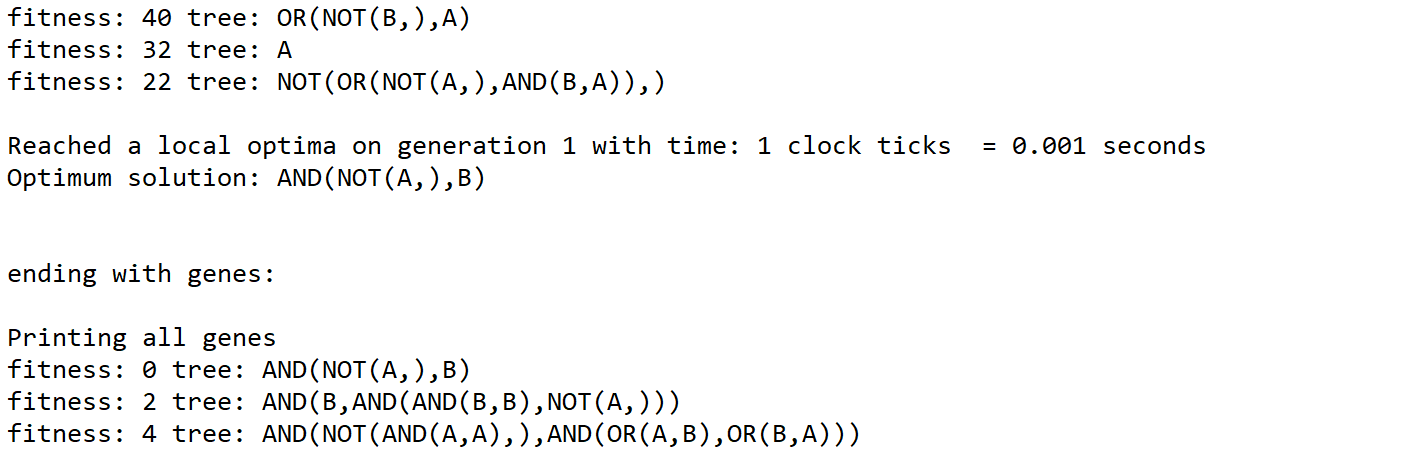
Description automatically generated



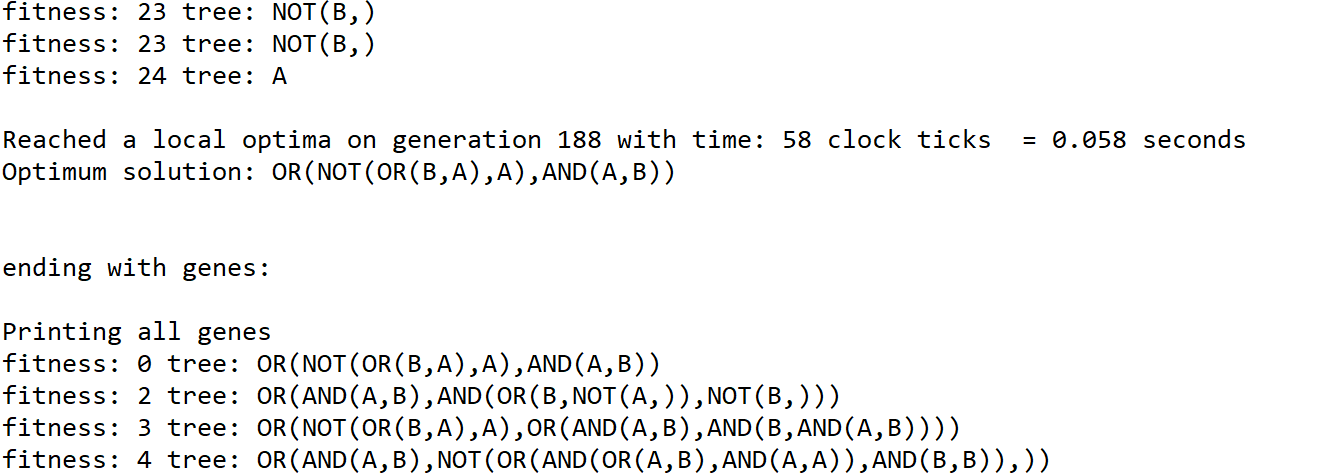
1. הוספנו תמיכה בפונקציה של AND(A,NOT(B))



AND(NOT(A),B)וגם את הפונקציה ההפוכה



וגם בפונקציית not xor:

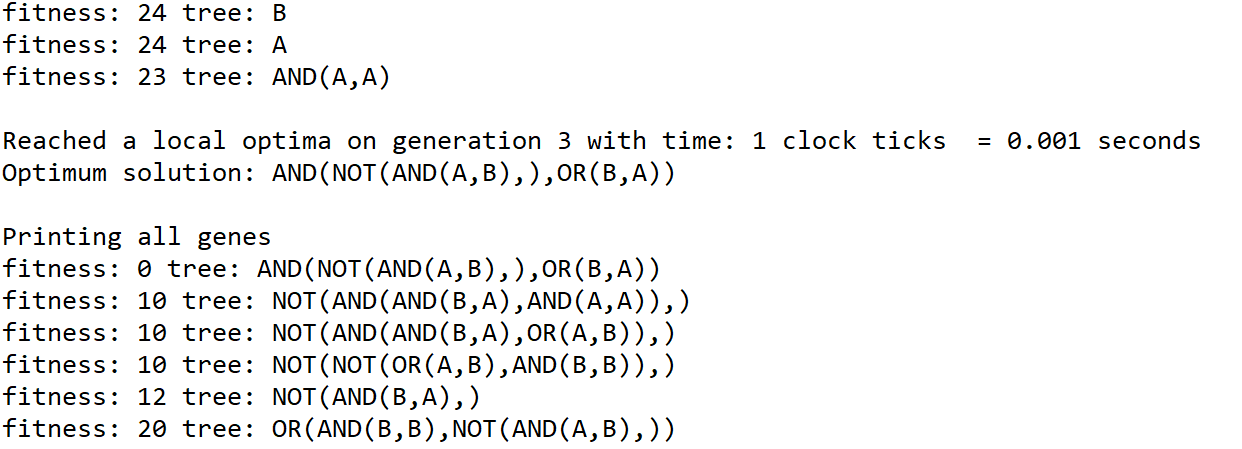


* ניתן להריץ קבצי exe מתאימים

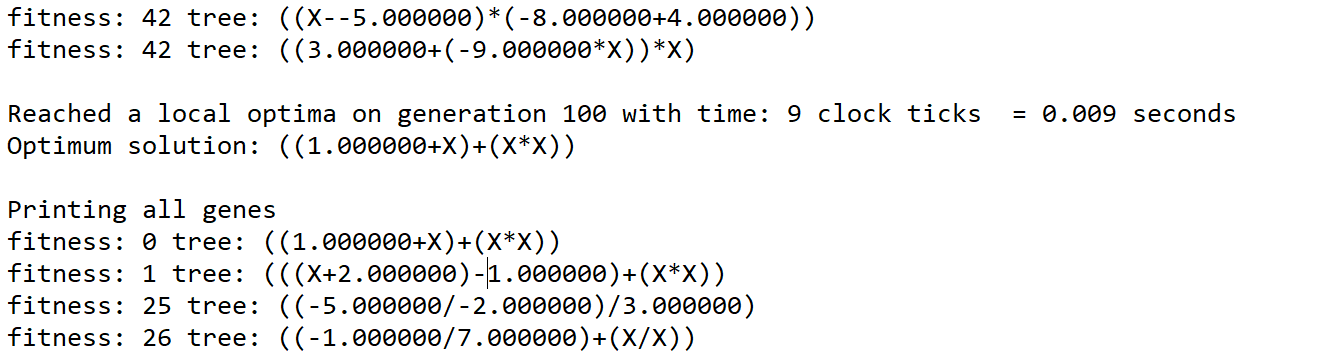
כדי ליעל את הבדיקה של אותו גן (עץ) עם ערכים שונים , החלטנו לנסות את כל הערכים האפשריים בו זמנית ולשמור אותם במערך. כלומר, לעבור רק פעם אחת על העץ ולחשב את תוצאות כל האופציות ביחד. עשינו זאת test\_tree\_all\_at\_onceבפונקציה

כדי ליעל את החישובים של אותם ערכים לגנים שונים לא מצאנו משהו שייעל את הבדיקה שם.

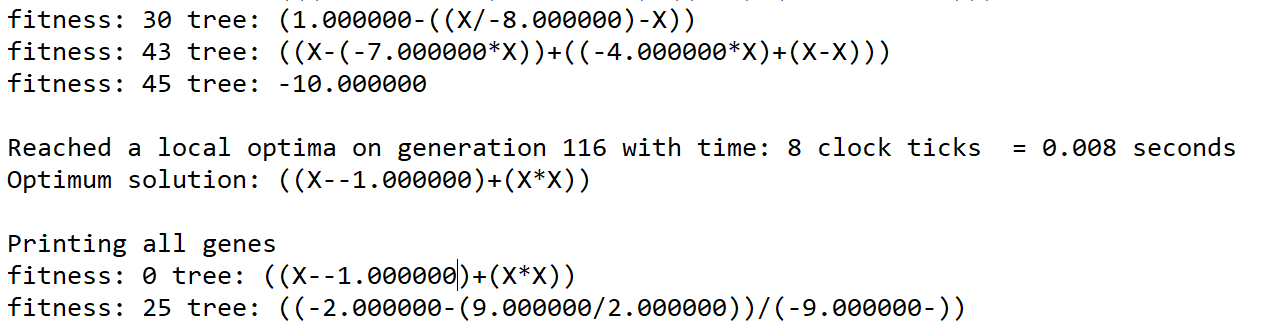
XORרואים בתמונה שזה טיפה יותר מהיר עבור פונקציית



1. הקוד כתוב , הינה תוצאות של הרצות:



ועוד הרצה



Part B:

for this part we chose to compare our results using knapsack problem from the previous lab. We set different number of threads for each test.

The results are summed in the tables below:

First table with two process:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test number\number of process | Normal time (one process) | Time | Speed up | Efficiency |
| 1 | 4.496s | 7.56s | 1.68 | 0.84 |
| 2 | 3.612s | 5.12s | 1.41 | 1.7 |
| 3 | 4.307s | 9.59s | 2.22 | 1.11 |
| 4 | 5.664s | 6.32s | 1.11 | 0.55 |
| 5 | 7.393s | 11.82s | 1.59 | 0.79 |
| 6 | 5.315s | 8.47s | 1.59 | 0.79 |
| 7 | 12.311s | 20.74s | 1.68 | 0.84 |
| 8 | 21.304s | 35.91s | 1.68 | 0.84 |

Second table with four process:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test number\number of process | Normal time (one process) | Time | Speed up | Efficiency |
| 1 | 4.496s | 9.32s | 2.07 | 0.51 |
| 2 | 3.612s | 8.65s | 2.39 | 0.59 |
| 3 | 4.307s | 12.74s | 2.957 | 0.73 |
| 4 | 5.664s | 10.37s | 1.83 | 0.45 |
| 5 | 7.393s | 15.20s | 2.055 | 0.51 |
| 6 | 5.315s | 11.03s | 2.07 | 0.51 |
| 7 | 12.311s | 29.83s | 2.423 | 0.60 |
| 8 | 21.304s | 51.77s | 2.43 | 0.60 |

\* All the test runs got to the optimal solution in the given time, so there was no need to compare the solutions.

Summary: out program did not improve with parallel computation; we think that the process creation took we too long which made our program slow.

We tried to increase the population size which helped with the parallel computation but got less positive results with our normal program (signal process).

We think that if we improve our process creation code or had a different problem which needed more time, the parallel computation would be way more effective.