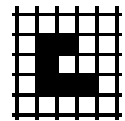


שאלה 2:

להרצת הקוד יש לוודא שהקבצים main.py, app.py, genetic_algo.py, matrix.py בנוסף צריך להתקין את החבילות pandas,numpy,tkinter.

ב. הסבירו בכתב את מימוש האלגוריתם – מהו הייצוג, מהם האופרטורים הגנטיים וכו'.

ייצוג פתרון: כל פתרון של מתושלח מוצלח מתואר כמערך של 0 ו 1 כאשר 0 זה תא ריק ו 1 זה תא חי. על מנת לפשט את האלגוריתם השתמשתי בכרומוזום (מערך) בגודל 9 המיצג מטריצה 3X3. לדוגמא הקונפיגורציה הבאה:



מיוצגת ככרומוזום: [1,1,1,0,1,0,0,1].

אוכלוסייה ראשונית: נבחר באקראי 10 כרומוזומים.

לכל כרומוזום נריץ את משחק החיים כדי לבדוק קונפיגורציה מקסימלית ונחזיר הקונפיגורציה כערך הכשירות של כרומוזום זה. לאחר שמצאנו את ערך הכשירות של כל כרומוזום נשמור את הכרומוזום בעל הכשירות הגבוה ביותר.

מודל ההתפתחות של האוכלוסייה:

- שלב הסלקציה: בחירת שני הורים לפי סלקציית רולטה.
- שלב השחלוף: מכל הורה נבחר תא באופן שווה (Uniform crossover).
- שלב המוטציה: בסיכוי של 50% כל תא יעבור שינוי (0 ל-1 ו 1 ל-0).
- נעבור בלולאה 10 פעמים ונוסיף כרומוזומים אלה לאוכלוסייה הבא.
- נעצור תהליך זה במידה ובמשך 5 דורות אין שיפור בפתרון הכי טוב או אחרי 20 דורות

פרמטרים בהם השתמשתי:

BOUNDING_BOX_AREA – גודל הכרומוזום
GAME_OF_LIFE_MAX_ITERATION – מספר האיטרציות המקסימלי של משחק החיים
MAT_SIZE – גודל הקונפיגורציה המרבית (בריבוע)
CELL_CIZE – גודל התא, נועד לממשק משתמש בלבד
POPULATION_SIZE – גודל אוכלוסייה
PROBABILITY_OF_LIVE – הסתברות לתא חי כשמאתחלים את הכרומוזומים הראשוניים.
MUTATION_PROBABILITY – הסתברות למוטציה לכל תא
MAX_GENERATIONS – מספר מקסימלי של דורות
MAX_GENERATIONS_WITHOUT_IMPROVMENT – מספר דורות ללא שינוי בפתרון

הערה: מצאתי פרמטרים אלה לפי ניסוי וטעיה.

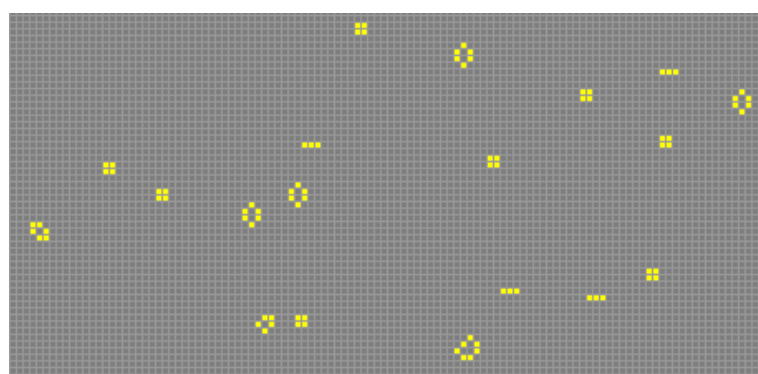
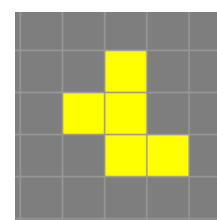
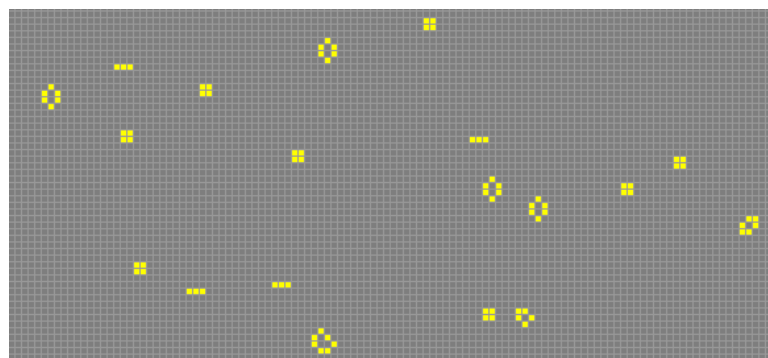
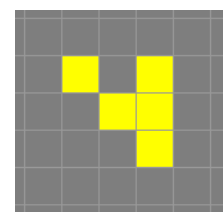
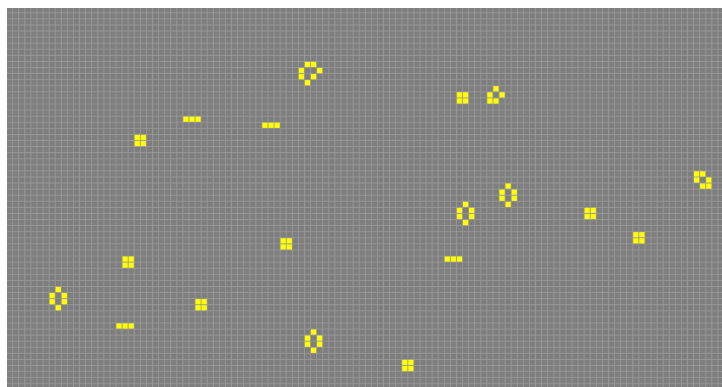
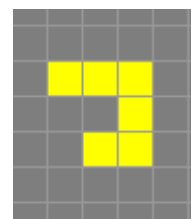
איך מוצאים כשירות לכל כרומוזום:

נשמור היסטוריה של שלושת המטריצות קודמות במהלך ריצת משחק החיים. במידה והמטריצה הבאה בריצה שקולה לאחת מהמטריצות מההיסטוריה זה אומר שיש מחזוריות. כאשר יש מחזוריות נחשב את השטח המקסימלי של התאים החיים של המטריצה הנוכחית וזה יהיה ערך הכשירות. אחרת נחזיר 0.

ג. הציגו את המתושלחים המוצלחים שמצאתם, ואת ההתפתחות שלהם במשחק החיים.

סעיף ג':

להלן "מתושלחים" מוצלחים שמצאתי:

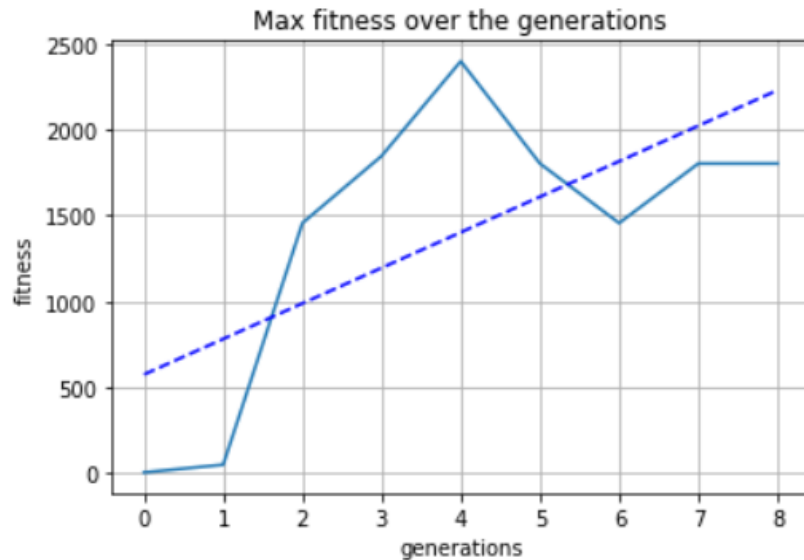


ד. נתחו את פעולת האלגוריתם הגנטי.
למשל:

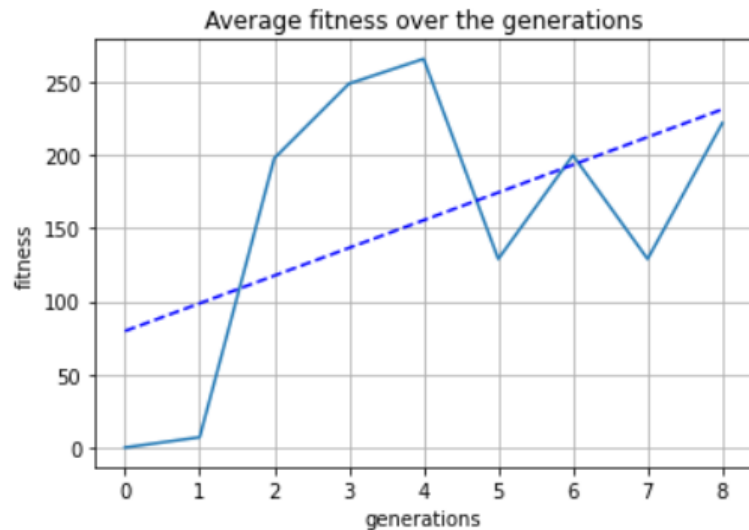
- ציירו גרף של הכשירות לאורך הדורות.
- הציגו פתרונות ביניים.
- חפשו נקודות זמן בהן הכשירות עלתה באופן משמעותי ונסו להסביר את הסיבה לכך.

סעיף ד':

גרף הכשירות לאורך הדורות כאשר נחשב בכל דור את הכשירות המקסימלית של האוכלוסייה:



גרף הכשירות לאורך הדורות כאשר נחשב בכל דור את הכשירות הממוצעת של האוכלוסייה:



על פי הגרפים, ניתן להבחין שבדור הראשון והשני הכשירות עלתה באופן משמעותי. הסיבה לכך היא שמספר האפשרויות לפתרון הוא $2^9=512$ (בהנחה שגודל הכרומוזום הוא 9). מספר זה הוא יחסית קטן והאלגוריתם הגנטי עם אוכלוסייה בגודל 10 יכול להגיע מאוד מהר לפתרון.

על מנת לראות פתרונות ביניים יש להריץ את התוכנית ולראות בפלט התוכנית את הפתרונות. לדוגמא:

```
found new temporary solution: [1, 1, 1, 1, 1, 0, 1, 1, 1]
```

```
found new temporary solution: [0, 1, 0, 0, 1, 1, 1, 1, 0]
```

```
found new temporary solution: [1, 1, 0, 0, 1, 1, 0, 1, 0]
```

```
found new temporary solution: [1, 1, 1, 1, 0, 1, 1, 0, 0]
```