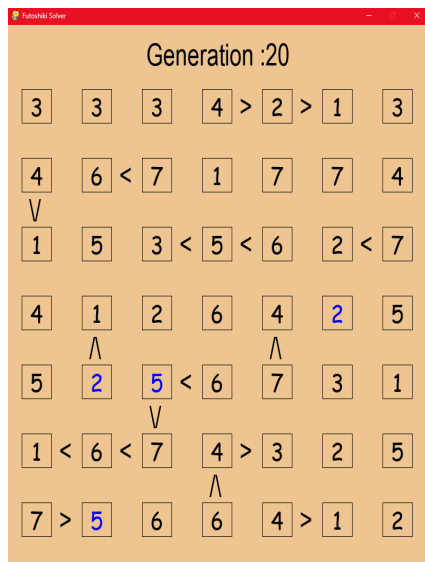


עבודה מס 2

תומר שורץ 209380864 (ביולוגיה חישובית)

בנוס של שילוב של ביולוגיה חישובית ומדעי המחשב

ייצוג UI של הפותר שלנו



ייצוג הפתרונות:

למשל וקטור ספרות שממנו מורכב כל פתרון חוקי אפשרי, למשל לוח 5X5 והספרות 4 ו2 נתונות על הלוח, כל הפתרונות יורכבו מ23 הספרות הבאות:

[1,1,1,1,1,2,2,2,2,3,3,3,3,3,4,4,4,4,5,5,5,5,5]

כלומר ישנם את כל הספרות 1-5 חמש פעמים ואנו רשמנו וקטור חמש פעמים
את הספרות 1,3,5 ואת הספרות 2 ו4 שמופיעות 4 פעמים בלוח כיוון שכבר
קיים להם מופע אחד.

פונקציית הערכה:

אנו מימשנו את פונקצית ההערכה - היא למעשה פונקציה שהגדרנו שכול
שאנו מקיימים יותר תנאים,
הציון שנקבל בפונקציה יהיה גבוה יותר.

אנו הגדרנו ציון מקסימלי של 100 לפתרון שעונה על כל הדרישות וחילקנו את סעיפי הניקוד של הפונקציה לשלושה חלקים:

1. על כמה תנאי "גדול מ" הפתרון עונה - מקסימום ניקוד בסעיף זה 70 נקודות
2. על כמה שורות תקינות קיימות בפתרון (כמה שורות אין חזרה של ספרות) - מקסימום ניקוד לסעיף זה 15 נקודות.
3. על כמה עמודות תקינות קיימות בפיתרון (כמה שורות אין חזרה של ספרות) - מקסימום ניקוד לסעיף זה 15 נקודות.

הסיבה שכך חילקנו את הנקודות, היא כיוון שרצינו לקבל ציון עגול שיהיה קל להבחין בהתקדמות איתו, והקפדנו לתת דגש גדול יותר לחלק הראשון כיוון שראינו שכאשר אנו נותנים ניקוד שווה לכל החלקים הפתרון מגיע לפתרון לא תקין אשר מקפיד על שתי החלקים האחרונים (כלומר 2,3) אך הוא לא מקפיד על תנאי 1. ולכן כאשר אנו נותנים דגש גדול יותר לחלק הראשון הפותר יקפיד יותר על החלק הראשון וגורם להגיע בקלות לפתרון אשר מקיים את כלל התנאים.

פונקציית ההכלאה:

אנו מימשנו את פונקצית ההכלאה בצורה הבאה:

לכל פתרון ניתן סיכוי בהתאם לציון פיטנאס שקיבל (שפירטנו קודם על פי מה חישבנו) כך שכלל שהציון היה יותר גבוהה כך גם הסיכוי יותר גבוהה

```
self.parent_percent = lambda x, min_fitness, max_fitness: (((x - min_fitness) / (max_fitness - min_fitness)) * .5) if max_fitness != min_fitness else 1 / 100
```

אחרי שנתנו לכל פתרון סיכוי להיבחר עברנו בלולאה כמות משתנה של פעמים (הסבר על זה בהמשך) שבכל איטרציה בחרנו באופן רנדומלי שתי פתרונות כך שלכל פתרון היה סיכוי להיבחר לפי מה שכתבנו הרגע. אחרי שבחרנו שתי פתרונות הגרלנו אינדקס רנדומלי והגרלנו פתרון מתחיל רנדומלי, ואז יצרנו פתרון חדש. הפתרון החדש הורכב מהחלק הראשון (אינדקס 0 עד האינדקס הרנדומלי) שלקוח כולו מהפתרון המתחיל שקבענו רנדומלית והחלק השני (מהאינדקס הרנדומלי עד לסוף הפתרון) שלקוח כולו מהפתרון השני שלא התחיל.

אחרי שנוצר הפתרון עברנו עליו עם פונקצית עזר שבודקת את התקינות שלו כדי לוודא שאין שם ספרות מיותרות/חסרות ולכל ספרה מיותרת החלפנו בספרה חסרה (זה יכול להיגרם בגלל שאנחנו לוקחים ספרות משתי פתרונות שונים שהספרות מחולקות אחרת בכל פתרון).

פונקצית מוטציות:

אנו מבצעים את המוטציות על ידי בחירת פתרון רנדומלי מתוך כלל הפתרונות, ועל פתרון זה אנו מבצעים פעולת מוטציה.

בכל תהליך של מוטציה אנחנו עושים כמות מסוימת של מוטציות (תלויה בגודל הפתרון והאם הגענו להתכנסות מוקדמת) ובכל מוטציה אנחנו בוחרים כל פעם 2 אינדקסים בצורה רנדומלית מתוך הפתרון ואנו מחליפים ביניהם. למעשה אנחנו בוחרים פתרון ומבצעים בו כמות מסוימת של החלפת ספרות בתוכו.

בעיית התכנסות מוקדמת:

אנו התמודדנו עם בעיית ההתכנסות המוקדמת תחילה על ידי זיהוי שאנו במצב זה, זיהוי זה נעשה בצורה הבאה:

בכל דור אנחנו בודקים אם הפתרונות הכי טוב נמצא לנו במילון של הפתרונות הטובים שאנחנו שומרים. אם הוא לא נמצא, נוסיף אותו לשם ובתור ערך נשים את מספר הפעמים שקיבלנו אותו עד עכשיו (1) ונשים בנוסף טאפל שיכיל את הפתרון ואת הציון שלו (בשביל השחזור בסוף הריצה). אם הפתרון נמצא במילון אנחנו נבדוק כמה פעמים הוא הופיע עד עכשיו ונוסיף לערך הזה 1. אם כמות הפעמים שהוא הופיע גדולה יותר מהכמות שהחלטנו (משתנה גלובלי שאפשר לשנות - הצבנו 25) אז המערכת עוברת למצב של התכנסות מוקדמת (לפעמים נקרא למצב זה גם , מקסימום לוקאלי) במשך כמה דורות (משתנה גלובלי שאפשר לשנות - הצבנו 5) לאחר שאנו יודעים שאנחנו במצב של התכנסות מוקדמת אנו למעשה מבצעים 2 דברים מרכזיים:

1. אנחנו משנים את סוג הפתרונות שאנחנו מקבלים בדור הבא (הסבר על זה בהמשך)
2. מגדילים את כמות המוטציות לפי 3 ממה שקיים כעת.

בחירת הדור הבא:

אנו החלטנו ליצור את הדור הבא כך שיש לנו 4 קטגוריות:

1. פתרונות שנוצרים כתוצאה מהכלאה של שני פתרונות (כמו שתיארנו למעלה)
2. פתרונות שנוצרו כתוצאה ממוטציות אקראיות בפתרונות רנדומלים מהדור הקודם (כמו שהסברנו למעלה)
3. הפתרונות הכי טובים שהיו בדור הקודם
4. פתרונות רנדומלים וחדשים לחלוטין.

כמות הפתרונות מכל קטגוריה משתנה בהתאם למצב התוכנית. אם התוכנית במצב רגיל אז החלוקה תהיה: 10%-40% אחוז פתרונות כתוצאה מהכלאה, 10%-40% פתרונות כתוצאה ממוטציות, 20% פתרונות מהדור הקודם, וכמות הפתרונות שנותר כדי להשלים לגודל האוכלוסיה שצריך (באופן רגיל להשלים 100 פתרונות) יהיו פתרונות רנדומליים חדשים. אם התוכנית במצב של מקסימום לוקאלי אז החלוקה תהיה: 45% פתרונות כתוצאה מהכלאה , 50% פתרונות כתוצאה ממוטציות ו5% מפתרונות רנדומלים חדשים.

כמות הדורות שהפותר רץ:

החלטנו להגביל את הפותר שלנו ל5,000 דורות כי ראינו שבדרך מאותו הזמן כבר רוב הפתרונות נמצאים ומאותו זמן ואילך הפותר לא מצליח להשתפר באמת.

השוואה בין הפותרים

בתרגיל נדרשנו לממש 3 סוגי פותרים שונים ולהשוות ביניהם:

פותר רגיל - מעריך את הציון פיטנאס של כל הפתרונות ומזה מכין את הפתרונות של הדור הבא
פותר דארווין - מעריך את הציון פיטנאס של כל הפתרונות אחרי שהם עברו אופטימיזציה אבל מכין את הפתרונות של הדור הבא מהפתרונות המקוריים

פותר לאמארקי - שמעריך את הציון פיטנאס של כל הפתרונות אחרי שהם עברו אופטימיזציה ומכין מהפתרונות שעברו אופטימיזציה את הפתרונות של הדור הבא.

את האופטימיזציה מימשנו באמצעות יצירת מוטציות נקודתיות ובדיקה האם הציון השתפר בעקבות האופטימיזציה. כדי לדעת איפה צריך לעשות מוטציה בפתרון יצרנו מטריצת עזר פתרון שהיתה מאותחלת להיות 0, ואז עברנו על הפתרון ובדקנו האם בכל שורה/עמודה יש בדיוק חזרה אחת של כל ספרה, בכל מקום שהיה לא תקין הוספנו לו ציון של 1 במטריצת עזר, בנוסף גם בדקנו על כל התנאים שניתנו לנו לראות אם כל המקומות שלהם מתקיימים, ומקום שלא קיים את התנאי קיבל תוספת של 10 במטריצת עזר. לבסוף המרנו את המטריצה בחזרה לזוקטור שיהיה מקביל לפתרון שלנו והשתמשנו בזוקטור הזה בשביל הסיכויים שבאינדקס מסוים בפתרון ילקח למוטציה.

פותר	מסקנה
<u>פותר רגיל</u>	אפשר לראות שסך הפתרונות משתפרים אבל הפתרון הכי טוב משתפר אבל באיטיות לעומת שאר הפותרים.
<u>פותר דארוויני</u>	אפשר לראות שהפותר משפר את ממוצע הפתרונות יותר מהר מהפותר הרגיל אבל הוא גם נכנס להרבה יותר למקסימומים לוקאליים.
<u>פותר לאמרק</u>	אפשר לראות שהפותר גורם לפתרונות לעלות בהרבה יותר מהירות לעומת הפתרונות האחרים.

כעת נסביר את הגרפים שאנו מפיקים:

ציר ה Y הוא הציון, ציר ה X זה הדורות

בכל גרף ציינו 4 קווים:

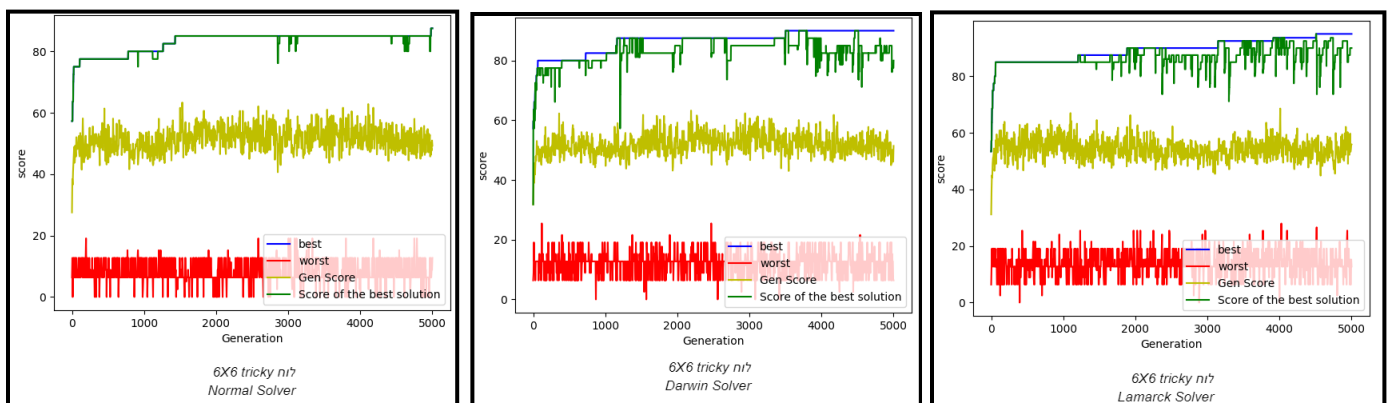
best - הפתרון הטוב ביותר שקיבלנו בכל הדורות.

gen score - הציון הממוצע של כל הפתרונות באותו דור. (כלומר לוקחים את הציונים של כל הפתרונות ומחלקים בכמות הפתרונות)

score of the best solution - הציון של הפתרון הכי טוב באותו דור.

worst - הפתרון הרע ביותר

כעת נציג את ההשוואות בין האלגוריתמים השונים בריצה על לוח 6X6 easy:



ניתן לראות שאנו מקבלים תוצאות בדיוק כפי שאנו ציינו (והסברנו על כך בעמודים הקודמים), כמו כן אפשר לראות שבכל האלגוריתמים אנו משפרים את אחוז הדיוק כל עוד אנו מבצעים יותר ויותר דורות. ניתן לראות שבכל שלושת הריצות הפתרון הממוצע קרוב יותר לפתרון הטוב ביותר מאשר לפתרון הרע.