

## ביולוגיה חישובית – תרגיל 3:

**הוראות הרצה** – קישור לגיט [פה](#). הרצת קובץ ההרצה עם ארגומנט ע"י הפקודה `som.exe Elec_24.csv`. בגיט נמצא גם קובץ פייתון שבו ניתן לראות את הקוד.

**חישוב המרחקים** – על מנת לחשב את המרחק בין ישוב מהקלט לבין תא בגריד השתמשנו במרחק אוקלידי. כלומר, בפונקציה `np.linalg.norm(x1 - x2)` של פייתון כך ש $x1$  הוא ישוב קלט  $x2$  הוא תא בגריד.

**אופן הקירוב** – ראשית, בדקנו מה ההפרש בין כל תא בוקטור הקלט לתא התואם לו בוקטור הגריד. עבור כל תא בוקטור הגריד החסרנו / הוספנו אליו 0.8 מההפרש שיצא כדי לצמצם את ההפרש שחישבנו בהתחלה.

**עדכון השכנים בדרגה ראשונה** – באותו האופן, ביצענו את חישוב ההפרש לכל תא. אך הפעם החסרנו / הוספנו לתא בוקטור הגריד 0.3 מההפרש שיצא.

**עדכון השכנים בדרגה השנייה** - באותו האופן, ביצענו את חישוב ההפרש לכל תא. אך הפעם החסרנו / הוספנו לתא בוקטור הגריד 0.1 מההפרש שיצא.

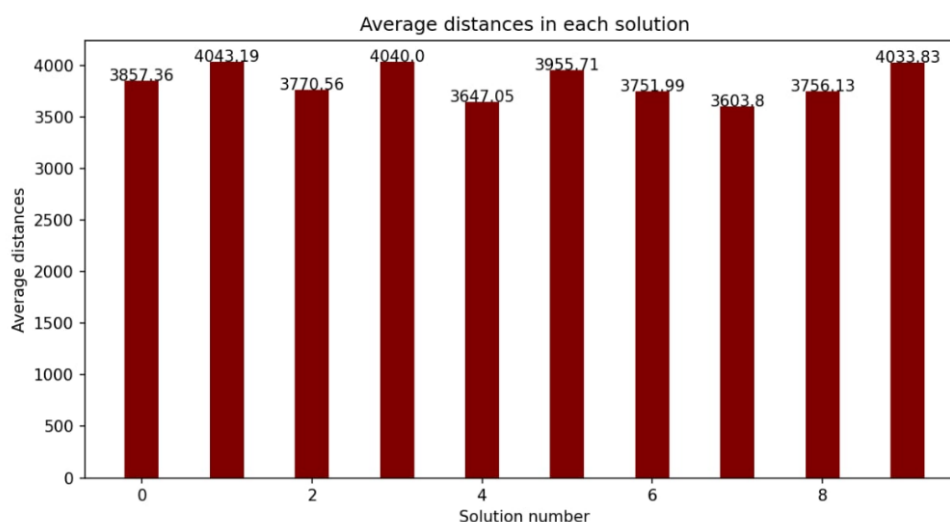
**אופן בחירת הצבע לתא בגריד** – במידה ולא מופו ישובים לתא מסוים נבצע אותו בצבע שחור.

לכל תא שמופו אליו ישובים ביצענו ממוצע של המצב הסוציאקונומי של הישובים שמופו אליו. לפי נתוני הקובץ, הממוצע יצא בין 0 ל-10. אתחלנו מערך של חמישה גוונים של כחול (המערך ממין מהכחול הבהיר ביותר לכהה ביותר). ככול שהכחול יותר בהיר מדובר בממוצע סוציאקונומי יותר נמוך. לאחר ביצוע הממוצע בדקנו את הטווח שבו הוא נמצא באופן הבא:

```
def get_color(avg):
    if avg == -1:
        return pygame.Color('black')
    if 0 <= avg <= 2:
        return colors[0]
    elif 2 < avg <= 4:
        return colors[1]
    elif 4 < avg <= 6:
        return colors[2]
    elif 6 < avg <= 8:
        return colors[3]
    elif 8 < avg <= 10:
        return colors[4]
```

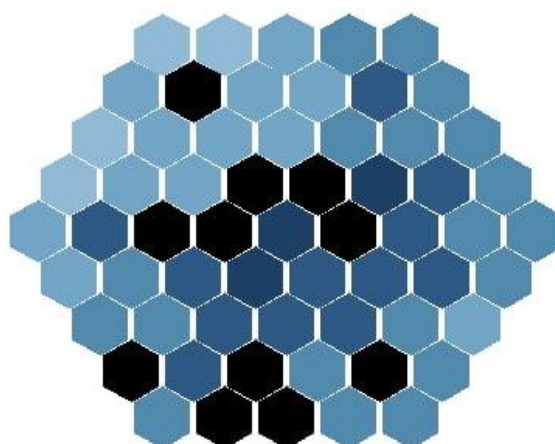
**בחירת פתרון טוב ביותר** – לאחר הרצת כל האיטרציות על רשת מסוימת, ביצענו חישוב של ממוצע המרחקים בין וקטור יישובי הקלט לבין וקטורי הרשת. ככל שהממוצע הנ"ל נמוך יותר, הפתרון טוב יותר כיוון שהצלחנו לקרב את וקטורי הקלט לווקטורי הרשת באופן הכי טוב.

נצרף תמונה של התוצאות לאחר ריצה ללא ערבוב וקטורי הקלט (כלומר, עשרה אתחולים שונים של רשתות רנדומיות, 10 איטרציות על כל רשת, ובחירה של פתרון טוב ביותר לפי הבדיקה שהסברנו לעיל):



כל עמודה מסמלת את הממוצע של ריצה מסוימת. הפתרון הטוב ביותר שהתקבל עבור הריצה הזו הוא של הרשת השמינית, כיוון שהעמודה של הרשת הזו הכי נמוכה.

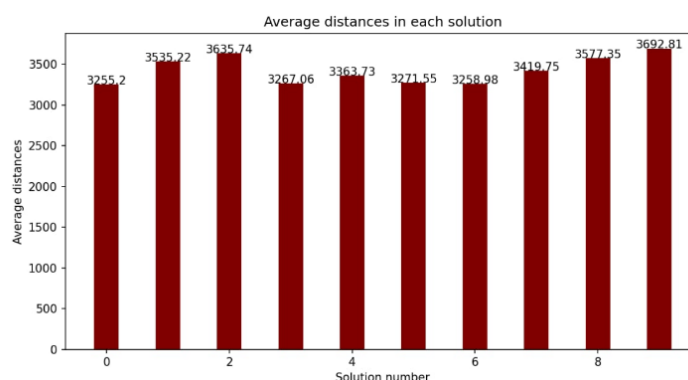
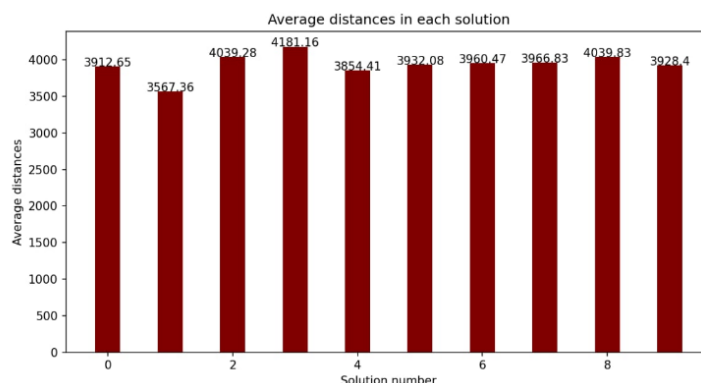
נצרף תמונה של הרשת הזו:



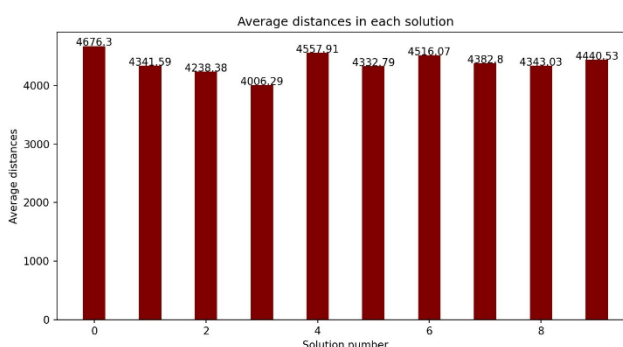
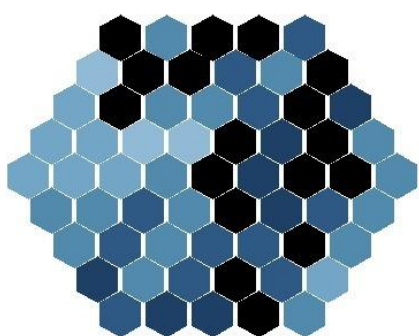
כפי שניתן לראות, הרשת די מחולקת ל-clusters. כלומר, התבצע קירוב של כל משושה ביחס לשכנים שלו וזו התוצאה שהתקבלה.

בעת סיום הריצה יודפס לטרמינל המיון של היישובים לפי המשושים (מיקום של משושה – ומערך של שמות היישובים שמוינו אליו). במידה ולא מוינו אליו ישובים יתקבל מערך ריק.

**שינוי סדר וקטורי הקלט** – ביצוע shuffle (כלומר, ערבוב רנדומי של וקטורי הקלט) לא משפיע על התוצאות בצורה משמעותית. נצרך תוצאות של הגרפים עבור ריצות שונות שוקטורי הקלט היו מעורבבים:



למעשה אין כל כך שינוי עבור ערבוב רנדומי. אך אם נמיין את הקלט לפי המצב הסוציאקונומי מהקטן לגדול נקבל את התוצאות הבאות:



ניתן לראות שהתקבלו תוצאות פחות טובות. ללא המיון טווח הממוצעים היה באזור 3000+ ולאחר המיון קיבלנו טווח ממוצעים של 4000+ ולכן התוצאה פחות טובה. גם ברשת ניתן לראות שהcluster'ים קצת פחות מסודרים לפי הצבעים וגם זה מצביע על כך שהפתרון פחות טוב.