Lab 4 Report

Samer Najjar ,ID:207477522

Morsy Biadsy ,ID:318241221

הסבר על מימוש שלנו:

במימוש שלנו לתרגיל עשינו בגדול שלושה מחלקות מחלקה ראשונה לאלגוריתם הגנתי שנייה לרשת המיון והשלישית לווקטורים שבודקים עליהם הרשתות שלנו

כדי לייצר את כל הווקטורים הבינאריים שאנחנו צריכים השתמשנו בקטע קוד הזה

```
def initVec(self):
    for i in range(2 ** self.N):
        bnumber = int(("{:0%db}" % self.N).format(i))  # create binary number
        bnumber = '{numbers:0{width}d}'.format(width=self.N, numbers=bnumber)
        member = SNVector(self.N, list(bnumber))
        self.vectors.append(member)
    print("VEC DONE!")
```

שעוברים בו על כל הערכים מ 0 על 2^k וכותבים בבינארי כאשר K הוא גודל המערך הנתון

לכל ווקטור מהווקטורים חישבנו FITNESS שלו שהוא היה כמה רשתות הווקטור הזה הכשיל

```python
def evaluate(self):
    sortingNetworks = 0
    for network in self.networks:
        sortingNetworks += self.checkSortingNetwork(network)
    self.fitness = len(self.networks) - sortingNetworks
    self.networks = []

def checkSortingNetwork(self, network):
    vec = []
    for i in range(len(self.vector)):
        vec.append(self.vector[i])
    for i in range(0, len(network.str), 2):
        i1 = vec[network.str[i]]
        i2 = vec[network.str[i + 1]]
        if i1 > i2:
            vec[network.str[i]] = '0'
            vec[network.str[i + 1]] = '1'
    for i in range(len(vec) - 1):
        if vec[i] > vec[i + 1]:
            return 0
    return 1
```

שכן בודקים אם הרשת ממיינת את הווקטור וככל שהוא מכשיל יותר רשתות יהי לו FITNESS יותר טובה

ולגבי רשת המיון עשינו דבר דומה חישבנו את הפיטנס

```python
def evaluate(self):
    vectors = 0
    for vec in self.vectors:
        vectors += self.checkSortingNetwork(vec)
    self.fitness = (len(self.vectors) - vectors) * 100 + len(self.str)

def checkSortingNetwork(self, vec):
    vecTemp = []
    for i in range(len(vec.vector)):
        vecTemp.append(vec.vector[i])

    for i in range(0, len(self.str), 2):
        i1 = vecTemp[self.str[i]]
        i2 = vecTemp[self.str[i + 1]]
        if int(i1) > int(i2):
            vecTemp[self.str[i]] = '0'
            vecTemp[self.str[i + 1]] = '1'
    for i in range(len(vecTemp) - 1):
        if vecTemp[i] > vecTemp[i + 1]:
            return 0
    return 1
```

בדקנו כמה ווקטורים הרשת הזו ממיינת ואז ככל שהיא ממיינת יותר ווקטורים מקבלת FITNESS טובה יותר וגם נתנו פקטור לרשתות שהאורך שלהם יותר קטן

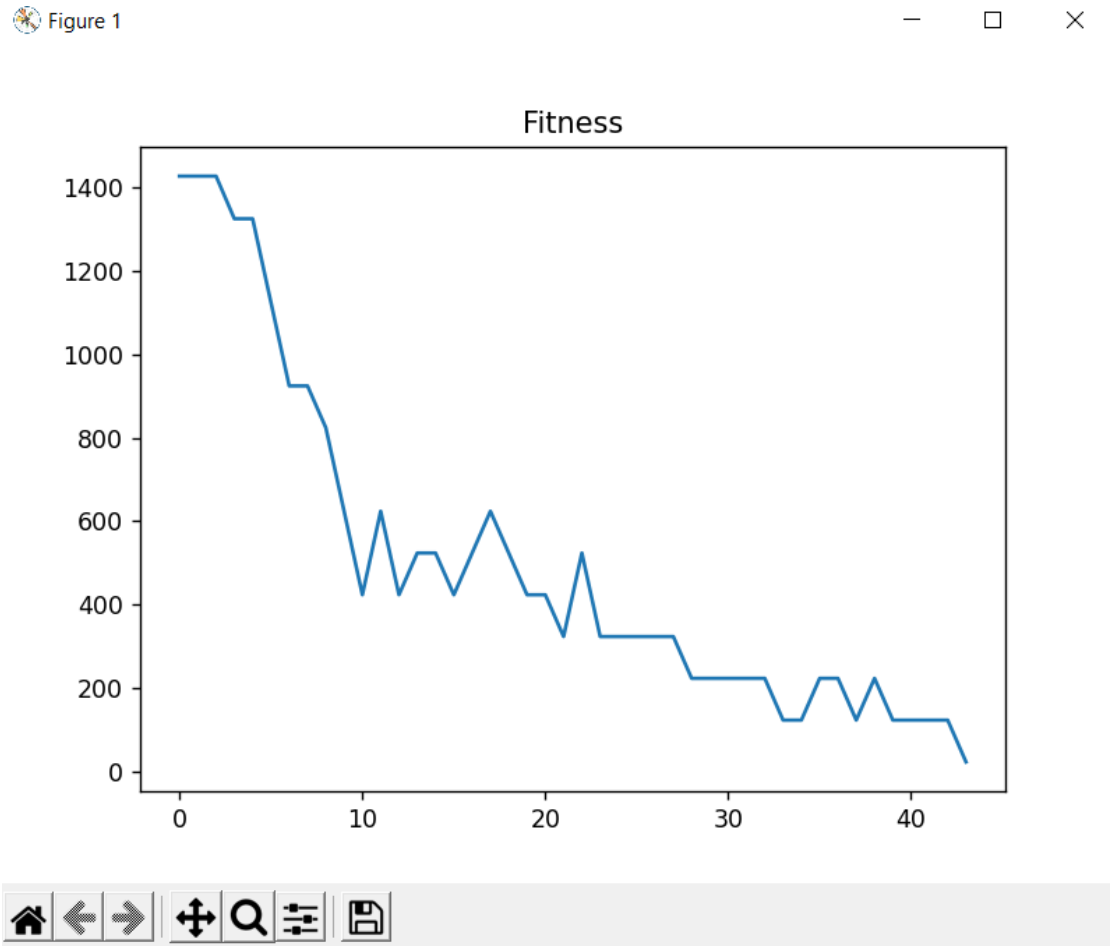
הערה: בגלל הפקטור של אורך המערך לא עשינו את הפיטנס להיות באחוזים ואז הפיטנס המינימאלית היא אורך הסדרה האופטימאלית

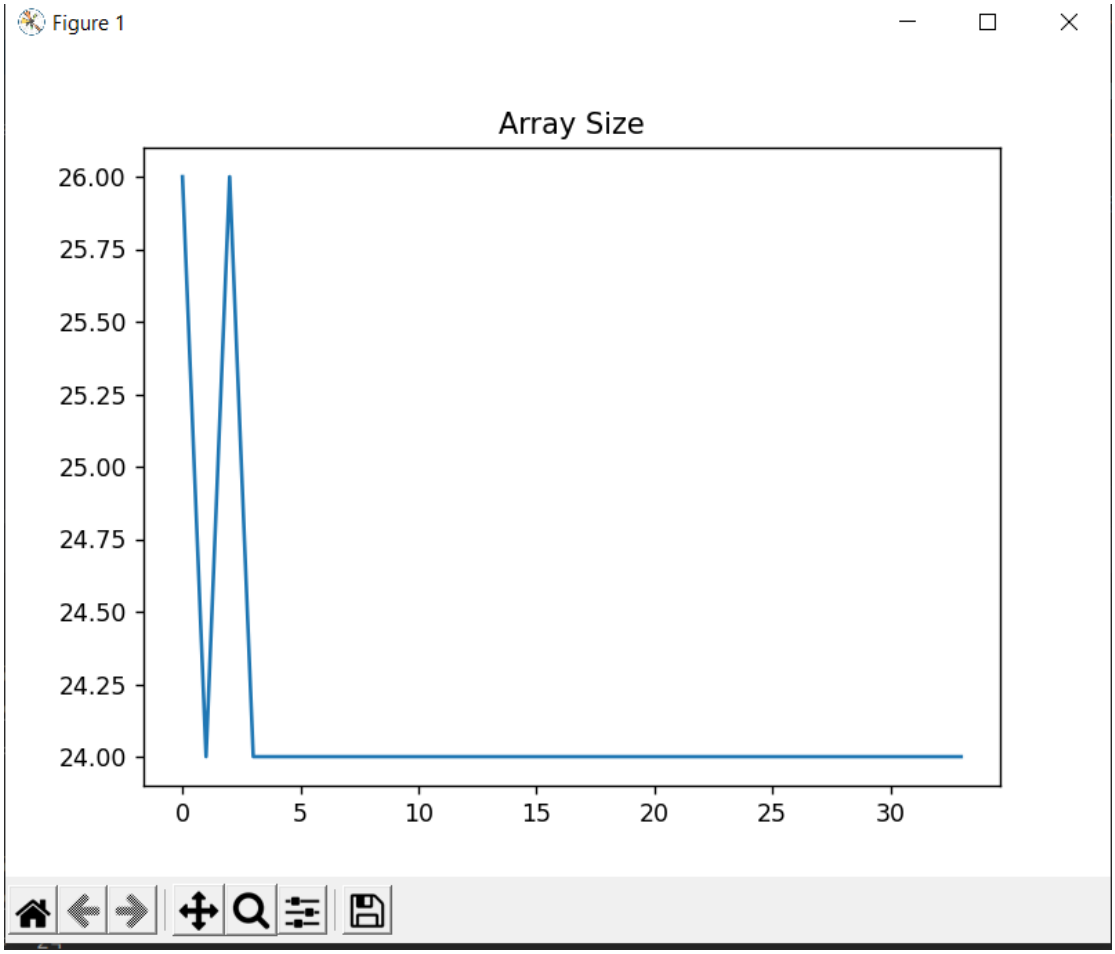
במקרה ש K=16 לא עברנו על כל הווקטורים (כי אז נסיים הרצה בסוף הסמסטר) ואז מה שעשינו זה שמיינו את הווקטורים לפי הפיטנס שלהם ואז בחרנו הווקטורים שהכשילו הרבה רשתות ובדקנו הרשתות החדשות עליהן אופן הבחירה היה כך:

```
def assign(self, itration):
    if self.N < 10:
        for i in range(self.popSize):
            self.population[i].vectors = self.vectors
            if i < len(self.vectors):
                self.vectors[i].networks = self.population
    else:
        if itration == 0:
            for i in range(self.popSize):
                self.population[i].vectors = []
                vectorsArr = sample(range(len(self.vectors)), k=GROUP_SIZE)
                for j in vectorsArr:
                    self.population[i].vectors.append(self.vectors[j])
                    self.vectors[j].networks.append(self.population[i])
        else:
            for i in range(self.popSize):
                self.population[i].vectors = []
                vectorsArr = sample(range(int(len(self.vectors) / 2)), k=GROUP_SIZE)
                for j in vectorsArr:
                    self.population[i].vectors.append(self.vectors[j])
                    self.vectors[j].networks.append(self.population[i])
```

דוגמא לריצה עבור K=6:

```
Run:  main
ITRATION :  31 / 5000
SOL =  [0, 5, 1, 2, 2, 4, 4, 5, 1, 2, 0, 1, 1, 3, 2, 4, 1, 2, 2, 3, 3, 5, 3, 4]
FITNESS =  224
ARRAY SIZE =  24

ITRATION :  32 / 5000
SOL =  [0, 5, 1, 2, 2, 4, 4, 5, 1, 2, 0, 1, 1, 3, 2, 4, 1, 2, 2, 3, 3, 5, 3, 4]
FITNESS =  224
ARRAY SIZE =  24

ITRATION :  33 / 5000
SOL =  [0, 5, 1, 2, 2, 4, 4, 5, 1, 2, 0, 1, 1, 3, 2, 4, 1, 2, 2, 3, 3, 5, 3, 4]
FITNESS =  224
ARRAY SIZE =  24

ITRATION :  34 / 5000
SOL =  [0, 5, 3, 2, 1, 4, 4, 5, 2, 5, 0, 1, 0, 3, 2, 4, 1, 3, 2, 3, 3, 4, 1, 2]
FITNESS =  24
ARRAY SIZE =  24


Process finished with exit code 0
```

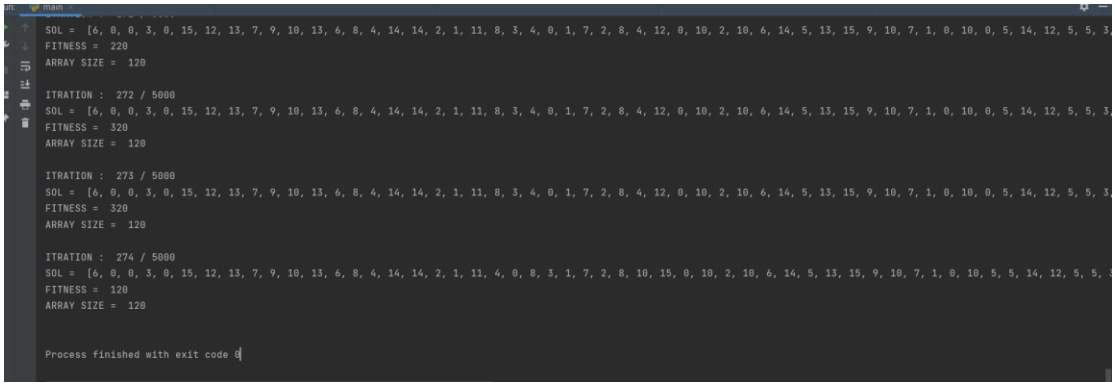


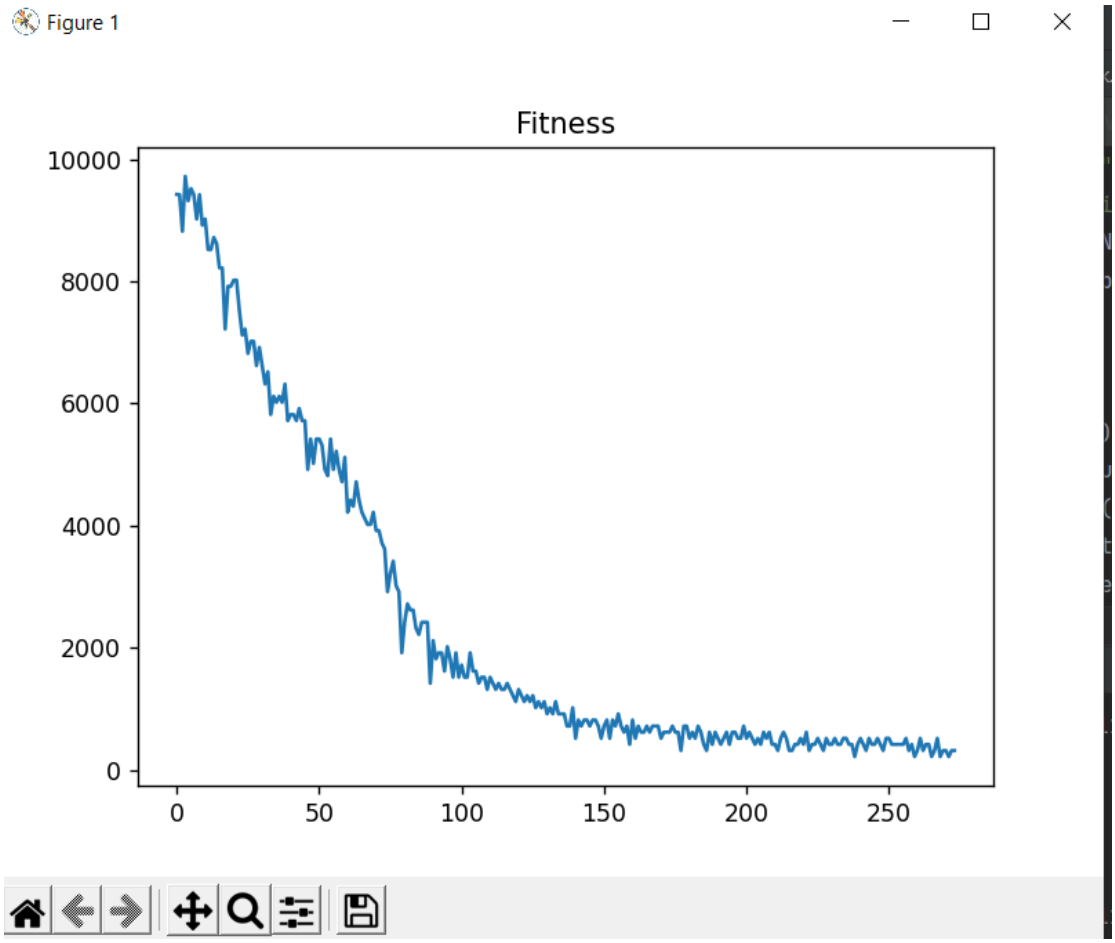


הסבר על הגרפים:

הגרף הראשון הוא גרף הפיטנס רואים שהוא משתפר עד שמגיע עד סדרה שממיינת כל מערך הגרף השני הוא גרף על גודל הסדרה עם הפיטנס הכי טובה בכל איטירציה

דוגמת ריצה עבור K=16:





הסדרה שקבלנו היא אינה ממיינת כל המערכים כי אנחנו בודקים רק על חלק קטן מהווקטורים אבל

זה הכי טוב שהאלגוריתם מצא כשבודקים על רק חלק מהסדרות