AI LAB 5

פייסל סעדיה 208336321

אחמד גבארין 314722307

שאלה 1:

כדי להגדיר את הסביבה שתתאים לבעיית חיפוש רשת אופטימלית ירשנו מ gym.env והוספנו כמה שדות ו override לכמה מתודות כמו , reset,step הוספנו שדה self.network שהוא instanceשל אותה מחלקה שבנינו עבור מעבדה 4 שהוספנו לה עוד דברים , הסוכן לומד את הפרמטר הזה שהוא בעצם מערך של מערכים בגודל 2 comparators .

אנחנו מלמדים את הרשת למיין 100 מערכים שזה מערך דו מימדי בגודל (100,num_elemnts) .

```python
class SortingNetworkEnv(gym.Env):
    def __init__(self, num_elements):
        super(SortingNetworkEnv, self).__init__()

        self.current_step=0
        self.added_comp=0
        self.num_elements = num_elements
        self.observation_space = spaces.Box(low=0, high=1, shape=(100,num_elements))  # State represents the current order of elements
        self.network = SortingNetwork(self.num_elements)

        self.state = np.random.rand(100, self.num_elements)  # Initialize the state with random order of elements
        # Update the action space to include swap, add, and remove operations
        self.action_space = spaces.Discrete(3)  # Three possible actions: swap, add, remove



    def reset(self):
        self.state = np.random.rand(100,self.num_elements)  # Initialize the state with random order of elements
        self.network = SortingNetwork(self.num_elements)
        self.current_step = 0
        self.added_comp=0
        return self.state.copy()
```

מימוש של המחלקה שמייצגת רשת מיון

```python
class SortingNetwork:
    def __init__(self, num_elements):
        self.num_elements = num_elements
        self.num_comparators = num_elements * (num_elements - 1) // 2
        self.generate_random()

    def copy(self):
        new_network = SortingNetwork(self.num_elements)
        new_network.comparators = np.copy(self.comparators)
        return new_network
    def generate_random(self):
        self.comparators = np.random.randint(0, self.num_elements, size=(self.num_comparators, 2))
        for comp in self.comparators:
            np.sort(comp)

    def sort(self, input_vector):
        for k in range(len(input_vector)):
            for i in range(len(self.comparators)):
```

שאלה 2:

```
# Create the SortingNetworkEnv
env = SortingNetworkEnv(num_elements=6)

...

model = DQN('MlpPolicy', env, verbose=1, learning_rate=0.1, gamma=0.95, exploration_fraction=0.3, buffer_size=1000)

# Train the agent
model.learn(total_timesteps=10000)

# # Save the trained model
# model.save("dqn_sorting_network")
```

שאלה 3:

הלימוד היה ע"י 3 פעולות בסיסיות הוספה \ מחיקה \ החלפה של איברים מהרשת , אתחלנו את גודל הרשת בגודל קבוע מראש תלוי בגודל מערך המטה בסדר גודל num_element^2 ואז הפעלנו את הפעולות במטרה להתכנס לרשת האופטימלית , הוספנו שדה חדש addedcomparator שהגדלנו אותו ב 1 בעת הוספת אעבר חדש לרשת (comparator) , והחסרנו אותו ב 1 כשהייתה פעולת מחיקה :

```
faisalsadi
def step(self, action):
    if self._is_sorted():
        return self.state.copy(),0,True,{}

    if action == 0:
        # Swap operation
        self.apply_swap_action()
        self.state=self.network.sort(self.state)
    elif action == 1:
        # Add operation
        self.apply_add_action()
        # if len(self.network.comparators) > 0:
        #     self.network.comparators[np.random.randint(0, len(self.network.compara
        self.state=self.network.sort(self.state)
    elif action == 2:
        # Remove operation
        self.apply_remove_action()
        self.state=self.network.sort(self.state)
    self.optimize()
    reward = -self._get_num_out_of_order_pairs() - self.added_comp
    self.current_step+=1
    # Check if the sorting is complete
    done = self._is_sorted() #or self.current_step>10000

    # Return the next state, reward, and done flag
    return self.state.copy(), reward, done, {}
```

פונקציית הוספת אלמנט לרשת מוסיפה אלמנט אקראי שמכיל השוואה בין שני אינדקסים אקראיים , כנל גם המחיקה מוחקת אלמנט אקראיי מהרשת וכך גם ההחלפה בוחרת שני אלמנטים אקראיים ברשת ומחליפה ביניהם

מודל התגמול היה שילוב בין מספר האיברים שנוספו לממוצע ה out_of_orders ב 100 המערכים כך :

```
self.optimize()
reward = -self._get_num_out_of_order_pairs() - self.added_comp
self.current_step+=1
# Check if the sorting is complete
done = self._is_sorted() #or self.current_step>10000
```

(a+b+c+d

```
# Create the SortingNetworkEnv
env = SortingNetworkEnv(num_elements=10)

...

model = DQN('MlpPolicy', env, verbose=1, learning_rate=0.1,gamma=0.95,exploration_fraction=0.3,buffer_size=1000)

# Train the agent
model.learn(total_timesteps=10000)
```

Learning_rate :

הגדרנו אותו להיות יחסית קטן , אומנם זה מאט את זמן האימון אך כדי להימנע מחוסר ודאות וזריקת פתרונות אופטימליים במהלך הלמדיה

Gamma:

קבענו אותו להיות גבוה קרוב ל 1 כדי לשאוף יותר טובות לתוצאות בעתיד טבע בעיית המיון דורש את הגישה הזאת

Buffer_size :

קבענו אותו כדי לאפשר לסוכן לצבור לזכור יותר מקרי עבר בתהליך הלמידה שזה אמור לספק לסוכן ניסיון בבעיה ואז אפשר להתכנס יותר טוב לרשת הנדרשת

Exploration_fatcor :

קבענו אותו לא גבוה מדי כדי שיהיה שילוב בין exploration ל exploitation עם קצת נטייה ל exploitation כי יש לנו 100 מערכים שמכסים הרבה מקרים אפשריים של פרמוטציות של מערכים לפחות עד 100 .

e)

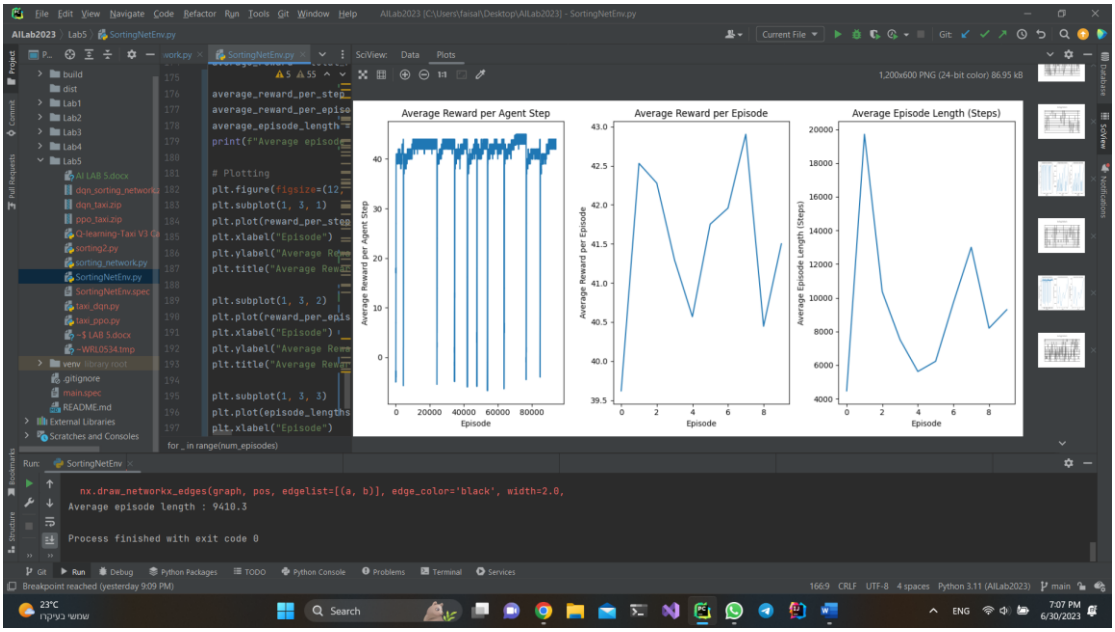
ו)

עבור num_elements=10 (מערך בגודל 10)

ו 10000 steps

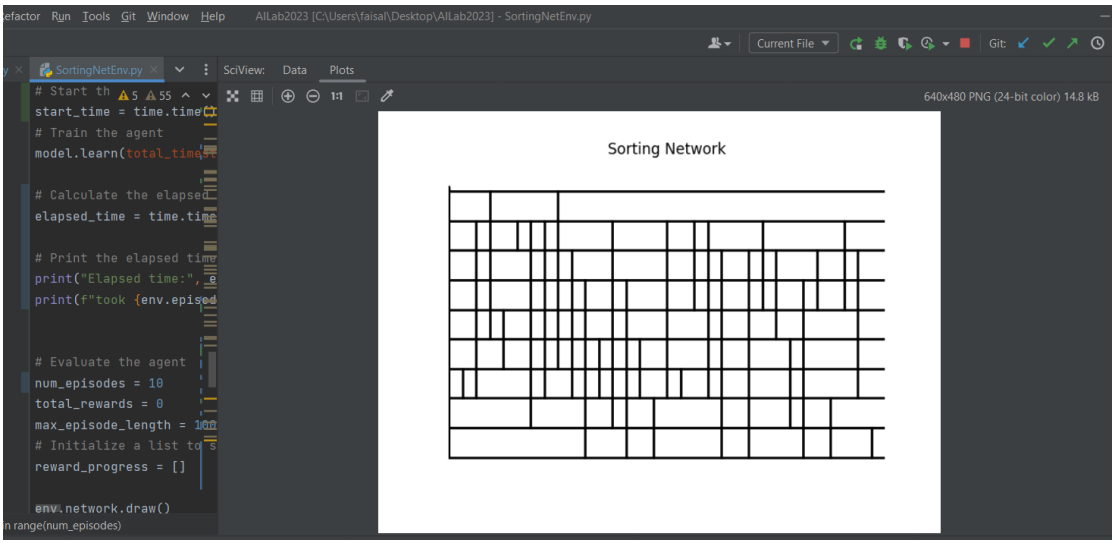
```
|    ep_rew_mean       | 9.82e+03 |
|    exploration_rate  | 0.05     |
| time/                |          |
|    episodes          | 12       |
|    fps               | 242      |
|    time_elapsed      | 28       |
|    total_timesteps   | 6903     |
-----------------------------------
-----------------------------------
| rollout/             |          |
|    ep_len_mean       | 557      |
|    ep_rew_mean       | 9.81e+03 |
|    exploration_rate  | 0.05     |
| time/                |          |
|    episodes          | 16       |
|    fps               | 248      |
|    time_elapsed      | 35       |
|    total_timesteps   | 8908     |
-----------------------------------
Elapsed time: 39.197532415390015 seconds
took 19 to learn the model
C:\Users\faisal\Desktop\AILab2023\Lab5\so
```

(II + III + IV

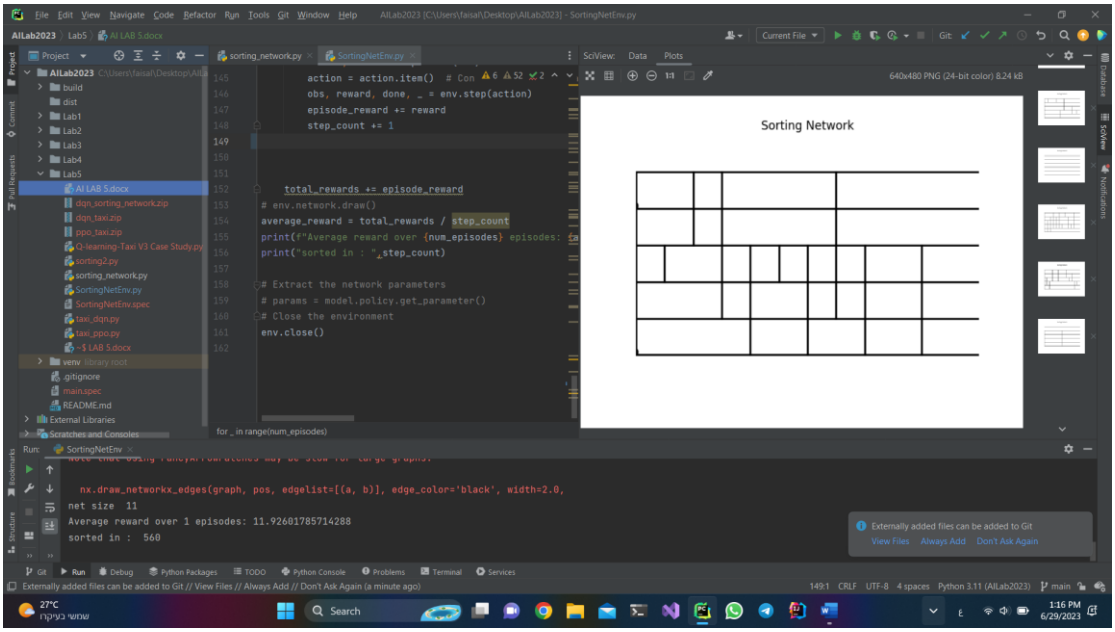


(V+VI+VII

רשת מיון עבור 10 איברים :



רשת עבור 6:


Sorting Network
net size 11
Average reward over 1 episodes: 11.92601785714288
sorted in : 560