RL Project

Aya Nageh 201901317
Riham Mohamed 201901639

About

State space:

Initial State:

Goal state:

Actions:

Apply:

Reward:

Assumptions:

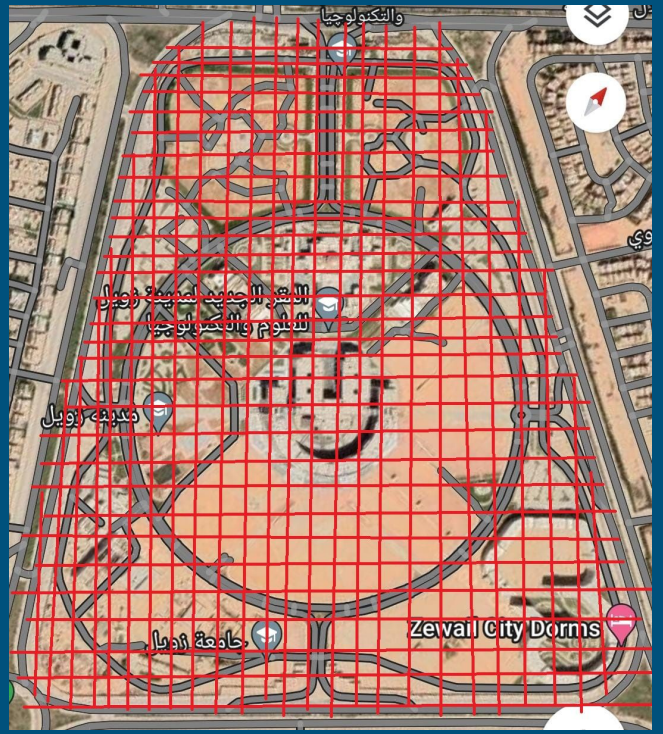
Strategy:

State Space

All the squares on the map, have 2D representation (row, column).

We will include item state, whether the agent has an item or not on the state to be 0 if no item, or 1 if it has item.

So the final representation will be (row, column, boolean:item)



Initial state

Main Gate is the initial state. The agent will reset to the main gate after each delivery.

Actions

1. We will give priority for pick up and drop off actions over the other actions (up, down, right, left, up_right, up_left, down_right, down_left)

2. Restriction: Check if the pickup locations have items or delivery locations have a vacancy before adding pick up or deliver actions.

Apply

Applies the action to the current state of the environment and returns the new state from applying the given action to the current environment state

Goal State:

The Agent will reset its position to Main gate, but will keep the q values

Assumptions

Put counter in pickup, and delivery locations with the number of items on them to be restriction before pick up or deliver .

There will be a variable indicate if the agent has item ‘to deliver’ or not on the state

Strategy:

We will work with exploring strategy to give the agent opportunity to explore the environment at the beginning and become more greedy at the end.

Reward

Rewards will be dependent on item, if the agent still do not have item, will gain negative reward. If it has an item will gain positive reward.

We will assign negative reward or relatively small rewards for needless movement to fasten the algorithm.

We will put relatively high negative reward on obstacles in order to save the agent form bad movements.

Reward

- Moving Through the Zewail City Borders: -5000 “Maximum negative reward”

- Moving Through the Desert: -500;

- Moving Through the Gardens: -200

- Moving Through the an Ordinary Road: -10;

- Moving Through any not goal Building (Dorms, Administration, Service, MainGate): -20

Reward

- Moving Through the Academic with an Item : +3000

- Moving Through the Academic without an Item: -3000

- Moving Through the OneStop with an Item : +3000

- Moving Through the OneStop without an Item : -300

Reward

- Moving Through the Nano without an Item : +3000

- Moving Through the Nano with an Item : -3000

- Moving Through the Helmy without an Item : +3000

- Moving Through the Helmy with an Item: -3000

Results

Find a Path?

Yes, in all trials.

We trying to train it for an hour and the results were better

Path and Policy:

```
q.items()
```


```
dict_items([(((39, 21), 'up'), 0.2620849889652055), (((38, 21), 'up'), -0.10082970913776784), (((37, 21), 'up'), -0.11621017388458821), (((36, 21), 'up'), -0.12353171374325456), (((35, 21), 'up'), -19999.999999998865), (((34, 21),
'up'), 0.43751921378054703), (((33, 21), 'up'), -19999.999999990905), (((32, 21), 'up'), -0.11617117502783003), (((31, 21), 'up'), 0.6357403122610776), (((30, 21), 'up'), 0.07004031607812067), (((29, 21), 'up'),
-19.99999999985448), (((28, 21), 'up'), -19.99999999985448), (((27, 21), 'up'), -19.99999999970896), (((26, 21), 'up'), -19.99999999970896), (((25, 21), 'up'), -19.999999999417923), (((24, 21), 'up'), -19.999999999417923), (((23,
21), 'up'), -0.4520701098663267), (((22, 21), 'up'), 157045.01521844068), (((21, 21), 'up'), 180470.0800547679), (((20, 21), 'up'), 12368.053687580395), (((19, 21), 'up'), 11838.009381957818), (((18, 21), 'up'),
12867.903107835446), (((17, 21), 'up'), 113633.97882309277), (((16, 21), 'up'), 28362.926242020912), (((15, 21), 'up'), 18780.993764633313), (((14, 21), 'up'), 173450.13489173725), (((13, 21), 'up'), 9687.025075033307), (((12,
21), 'up'), -19999.995231628418), (((11, 21), 'up'), -19999.999925494194), (((10, 21), 'up'), -19999.99998137355), (((9, 21), 'up'), -19999.999995343387), (((8, 21), 'up'), -19999.999995343387), (((7, 21), 'up'),
-19999.999997671694), (((6, 21), 'up'), -19999.999998835847), (((5, 21), 'up'), -19999.999999417923), (((4, 21), 'up'), -19999.99999970896), (((3, 21), 'up'), 0.11790743395249592), (((2, 21), 'up'), -19999.99999970896), (((1, 21),
'up'), -4999999999.92724), (((0, 21), 'up'), -4999999999.92724), (((0, 21), 'up-right'), -4999999999.92724), (((0, 22), 'up'), -4999999999.92724), (((0, 22), 'up-right'), -4999999999.92724), (((0, 23), 'up'), -4999999999.92724),
(((0, 23), 'up-right'), -4999999999.92724), (((0, 24), 'up'), -4999999999.92724), (((0, 24), 'up-right'), -4999999999.92724), (((0, 25), 'up'), -4999999999.92724), (((0, 25), 'up-right'), -4999999999.92724), (((0, 26), 'up'),
-4999999999.92724), (((0, 26), 'up-right'), -4999999999.92724), (((0, 27), 'up'), -4999999999.92724), (((0, 27), 'up-right'), -4999999999.92724), (((0, 28), 'up'), -4999999999.92724), (((0, 28), 'up-right'), -4999999999.92724),
(((0, 29), 'up'), -4999999999.92724), (((0, 29), 'up-right'), -4999999999.92724), (((0, 30), 'up'), -4999999999.96362), (((0, 30), 'up-right'), -4999999999.96362), (((0, 31), 'up'), -4999999999.96362), (((0, 31), 'up-right'),
-4999999999.96362), (((0, 32), 'up'), -4999999999.96362), (((0, 32), 'up-right'), -4999999999.96362), (((0, 33), 'up'), -4999999999.96362), (((0, 33), 'up-right'), -4999999999.96362), (((0, 34), 'up'), -4999999999.96362), (((0,
34), 'up-right'), -4999999999.96362), (((0, 35), 'up'), -4999999999.96362), (((0, 35), 'up-right'), -4999999999.96362), (((0, 36), 'up'), -4999999999.96362), (((0, 36), 'up-right'), -4999999999.96362), (((0, 37), 'up'),
-4999999999.96362), (((0, 37), 'up-right'), -4999999999.96362), (((0, 38), 'up'), -4999999999.96362), (((0, 38), 'up-right'), -4999999999.96362), (((0, 39), 'up'), -4999999999.96362), (((0, 39), 'up-right'), -4999999999.96362),
(((0, 40), 'up'), -4999999999.96362), (((0, 40), 'up-right'), -4999999999.96362), (((0, 41), 'up'), -4999999999.96362), (((0, 41), 'up-right'), -4999999999.96362), (((0, 41), 'up-left'), -4999999999.96362), (((0, 40), 'up-left'),
-4999999999.96362), (((0, 39), 'up-left'), -4999999999.96362), (((0, 38), 'up-left'), -4999999999.96362), (((0, 37), 'up-left'), -4999999999.96362), (((0, 36), 'up-left'), -4999999999.96362), (((0, 35), 'up-left'),
-4999999999.96362), (((0, 34), 'up-left'), -4999999999.96362), (((0, 33), 'up-left'), -4999999999.96362), (((0, 32), 'up-left'), -4999999999.96362), (((0, 31), 'up-left'), -4999999999.96362), (((0, 30), 'up-left'),
-4999999999.92724), (((0, 29), 'up-left'), -4999999999.92724), (((0, 28), 'up-left'), -4999999999.92724), (((0, 27), 'up-left'), -4999999999.92724), (((0, 26), 'up-left'), -4999999999.92724), (((0, 25), 'up-left'),
-4999999999.92724), (((0, 24), 'up-left'), -4999999999.92724), (((0, 23), 'up-left'), -4999999999.92724), (((0, 22), 'up-left'), -4999999999.92724), (((0, 21), 'up-left'), -4999999999.92724), (((0, 20), 'up'), -4999999999.92724),
(((0, 20), 'up-right'), -4999999999.92724), (((0, 21), 'down'), -19999.99999970896), (((1, 21), 'up-right'), -4999999999.92724), (((0, 22), 'down'), -19999.99999970896), (((1, 22), 'up'), -4999999999.92724), (((0, 22), 'down-
right'), -19999.99999970896), (((1, 23), 'up'), -4999999999.92724), (((0, 23), 'down'), -19999.99999970896), (((1, 23), 'up-right'), -4999999999.92724), (((0, 24), 'down'), -19999.99999970896), (((1, 24), 'up'),
-4999999999.92724), (((0, 24), 'down-right'), -19999.99999970896), (((1, 25), 'up'), -4999999999.92724), (((0, 25), 'down'), -19999.99999970896), (((1, 25), 'up-right'), -4999999999.92724), (((0, 26), 'down'),
-0.48672483936388744), (((1, 26), 'up'), -4999999999.92724), (((0, 26), 'down-right'), -0.48672483936388744), (((1, 27), 'up'), -4999999999.92724), (((0, 27), 'down'), -0.49844358936388744), (((1, 27), 'up-right'),
-4999999999.92724), (((0, 28), 'down'), -0.48818968311388744), (((1, 28), 'up'), -4999999999.92724), (((0, 28), 'down-right'), -0.48818968311388744), (((1, 29), 'up'), -4999999999.92724), (((0, 29), 'down'), -0.34756468311388744),
(((1, 29), 'up-right'), -4999999999.92724), (((0, 30), 'down'), -0.39443968311388744), (((1, 30), 'up'), -4999999999.92724), (((0, 30), 'down-right'), -0.39443968311388744), (((1, 31), 'up'), -4999999999.96362), (((0, 31),
'down'), 0.20903015844305628), (((1, 31), 'up-right'), -4999999999.92724), (((0, 32), 'down'), 0.11528015844305628), (((1, 32), 'up'), -4999999999.96362), (((0, 32), 'down-right'), 0.11528015844305628), (((1, 33), 'up'),
-4999999999.96362), (((0, 33), 'down'), 0.11528015844305628), (((1, 33), 'up-right'), -4999999999.96362), (((0, 34), 'down'), 0.11528015844305628), (((1, 34), 'up'), -4999999999.96362), (((0, 34), 'down-right'),
0.11528015844305628), (((1, 35), 'up'), -4999999999.96362), (((0, 35), 'down'), 0.11528015844305628), (((1, 35), 'up-right'), -4999999999.96362), (((0, 36), 'down'), 0.11528015844305628), (((1, 36), 'up'), -4999999999.96362),
(((0, 36), 'down-right'), 0.11528015844305628), (((1, 37), 'up'), -4999999999.96362), (((0, 37), 'down'), 0.11528015844305628), (((1, 37), 'up-right'), -4999999999.96362), (((0, 38), 'down'), 0.11528015844305628), (((1, 38),
'up'), -4999999999.96362), (((0, 38), 'down-right'), 0.11528015844305628), (((1, 39), 'up'), -4999999999.96362), (((0, 39), 'down'), 0.11528015844305628), (((1, 39), 'up-right'), -4999999999.96362), (((0, 40), 'down'),
0.11528015844305628), (((1, 40), 'up'), -4999999999.96362), (((0, 40), 'down-right'), 0.11528015844305628), (((1, 41), 'up'), -4999999999.96362), (((0, 41), 'down'), 0.11528015844305628), (((1, 41), 'up-right'),
-4999999999.96362), (((0, 41), 'down-right'), 0.11528015844305628), (((1, 41), 'up-left'), -4999999999.96362), (((0, 40), 'down-left'), 0.11528015844305628), (((1, 39), 'up-left'), -4999999999.96362), (((0, 38), 'down-left'),
0.11528015844305628), (((1, 37), 'up-left'), -4999999999.96362), (((0, 36), 'down-left'), 0.11528015844305628), (((1, 35), 'up-left'), -4999999999.96362), (((0, 34), 'down-left'), 0.11528015844305628), (((1, 33), 'up-left'),
-4999999999.96362), (((0, 32), 'down-left'), -0.39443968311388744), (((1, 31), 'up-left'), -4999999999.92724), (((0, 30), 'down-left'), -0.48818968311388744), (((1, 29), 'up-left'), -4999999999.92724), (((0, 28), 'down-left'),
-0.48672483936388744), (((1, 27), 'up-left'), -4999999999.92724), (((0, 26), 'down-left'), -19999.99999970896), (((1, 25), 'up-left'), -4999999999.92724), (((0, 24), 'down-left'), -19999.99999970896), (((1, 23), 'up-left'),
-4999999999.92724), (((0, 22), 'down-left'), -19999.99999970896), (((1, 21), 'up-left'), -4999999999.92724), (((0, 20), 'up-left'), -4999999999.92724), (((0, 19), 'up'), -4999999999.92724), (((0, 19), 'up-right'),
-4999999999.92724), (((0, 20), 'down'), 0.11797605379979359), (((1, 20), 'up'), -4999999999.92724), (((0, 20), 'down-right'), -19999.99999970896), (((1, 21), 'down'), 0.11797605379979359), (((2, 21), 'up-right'),
-19999.99999970896), (((1, 22), 'up-right'), -4999999999.92724), (((0, 23), 'down-right'), -19999.99999970896), (((1, 24), 'up-right'), -4999999999.92724), (((0, 25), 'down-right'), 0.24441529735486256), (((1, 26), 'up-right'),
-4999999999.92724), (((0, 27), 'down-right'), -0.49844358936388744), (((1, 28), 'up-right'), -4999999999.92724), (((0, 29), 'down-right'), -0.34756468311388744), (((1, 30), 'up-right'), -4999999999.92724), (((0, 31), 'down-
).96362), (((0, 33), 'down-right'), 0.11528015844305628), (((1, 34), 'up-right'), -4999999999.96362), (((0, 35), 'down-right'), 0.11528015844305628), (((1, 36), 'up-
```


matic saving failed. This file was updated remotely or in another tab. Show diff

The path is not Optimal as it needs more iterations for the agent to learn with learning rate 0.5.

It is hard for the agent to learn the optimal path as the requirement is a little bit complicated.