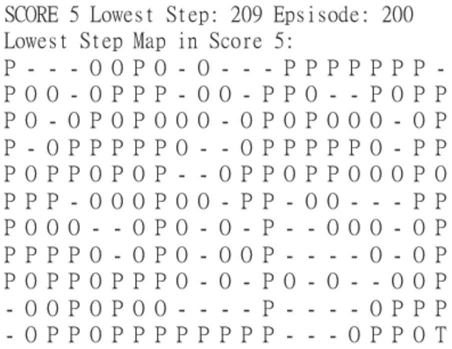
人工智慧與機器學習作業三 RL Maze　　　　資管三 B 109403533 林采璇

一、 Colab 連結

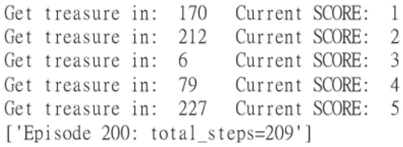
https://colab.research.google.com/drive/1iT2QLuC3Wks3SJgiLVQx27azgCr19MVS?usp=sharing

二、 最佳結果截圖

- 找到 5 個寶藏所需步數：209

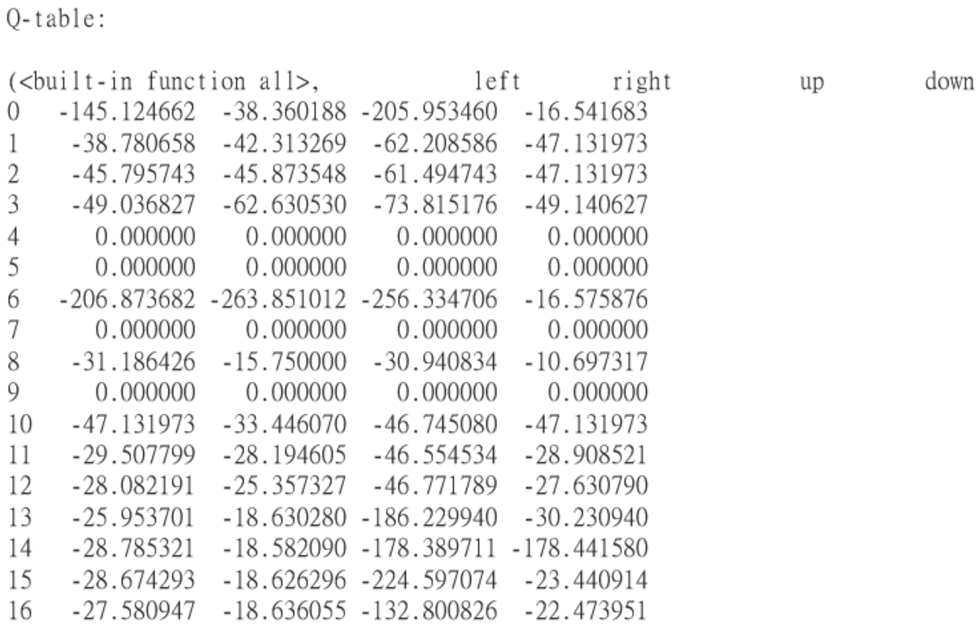


```
SCORE 5 Lowest Step: 209 Epsisode: 200
Lowest Step Map in Score 5:
P - - - O O P O - O - - - P P P P P P P -
P O O - O P P P - O O - P P O - - P O P P
P O - O P O P O O O - O P O P O O O - O P
P - O P P P P P O - - O P P P P P O - P P
P O P P O P O P - - O P P O P P O O O P O
P P P - O O O P O O - P P - O O - - - P P
P O O O - - O P O - O - P - - O O O - O P
P P P P O - O P O - O O P - - - - O - O P
P O P P O P P P O - O - P O - O - - O O P
- O O P O P O O - - - - P - - - - O P P P
- O P P O P P P P P P P P - - - O P P O T
```



```
Get treasure in:  170   Current SCORE:  1
Get treasure in:  212   Current SCORE:  2
Get treasure in:  6     Current SCORE:  3
Get treasure in:  79    Current SCORE:  4
Get treasure in:  227   Current SCORE:  5
['Episode 200: total_steps=209']
```

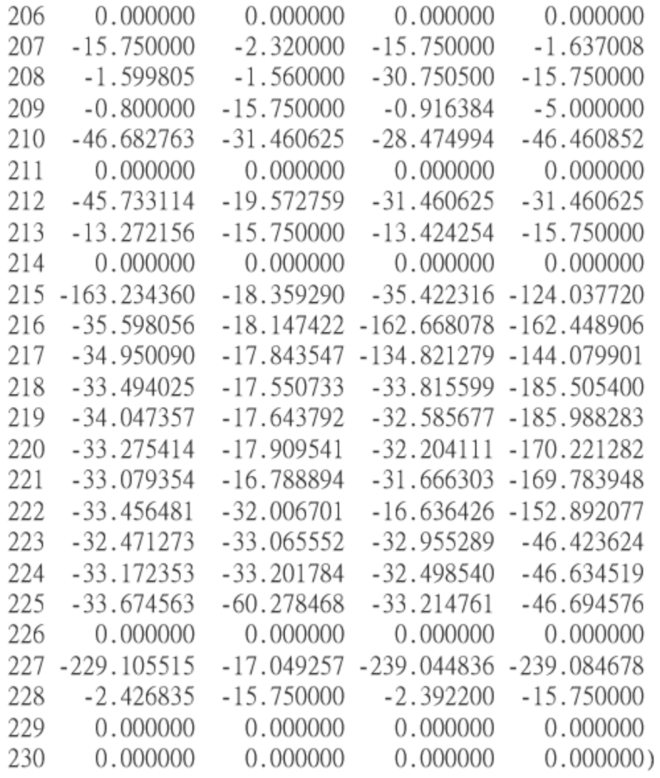
三、　Q-table 結果截圖



```
Q-table:

(<built-in function all>,            left        right           up          down
0    -145.124662   -38.360188 -205.953460   -16.541683
1     -38.780658   -42.313269   -62.208586   -47.131973
2     -45.795743   -45.873548   -61.494743   -47.131973
3     -49.036827   -62.630530   -73.815176   -49.140627
4       0.000000     0.000000     0.000000     0.000000
5       0.000000     0.000000     0.000000     0.000000
6    -206.873682 -263.851012 -256.334706   -16.575876
7       0.000000     0.000000     0.000000     0.000000
8     -31.186426   -15.750000   -30.940834   -10.697317
9       0.000000     0.000000     0.000000     0.000000
10    -47.131973   -33.446070   -46.745080   -47.131973
11    -29.507799   -28.194605   -46.554534   -28.908521
12    -28.082191   -25.357327   -46.771789   -27.630790
13    -25.953701   -18.630280 -186.229940   -30.230940
14    -28.785321   -18.582090 -178.389711 -178.441580
15    -28.674293   -18.626296 -224.597074   -23.440914
16    -27.580947   -18.636055 -132.800826   -22.473951
```

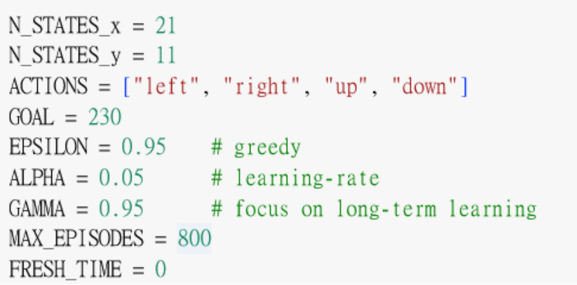
…



```
206     0.000000     0.000000     0.000000     0.000000
207   -15.750000    -2.320000   -15.750000    -1.637008
208    -1.599805    -1.560000   -30.750500   -15.750000
209    -0.800000   -15.750000    -0.916384    -5.000000
210   -46.682763   -31.460625   -28.474994   -46.460852
211     0.000000     0.000000     0.000000     0.000000
212   -45.733114   -19.572759   -31.460625   -31.460625
213   -13.272156   -15.750000   -13.424254   -15.750000
214     0.000000     0.000000     0.000000     0.000000
215  -163.234360   -18.359290   -35.422316  -124.037720
216   -35.598056   -18.147422  -162.668078  -162.448906
217   -34.950090   -17.843547  -134.821279  -144.079901
218   -33.494025   -17.550733   -33.815599  -185.505400
219   -34.047357   -17.643792   -32.585677  -185.988283
220   -33.275414   -17.909541   -32.204111  -170.221282
221   -33.079354   -16.788894   -31.666303  -169.783948
222   -33.456481   -32.006701   -16.636426  -152.892077
223   -32.471273   -33.065552   -32.955289   -46.423624
224   -33.172353   -33.201784   -32.498540   -46.634519
225   -33.674563   -60.278468   -33.214761   -46.694576
226     0.000000     0.000000     0.000000     0.000000
227  -229.105515   -17.049257  -239.044836  -239.084678
228    -2.426835   -15.750000    -2.392200   -15.750000
229     0.000000     0.000000     0.000000     0.000000
230     0.000000     0.000000     0.000000     0.000000)
```

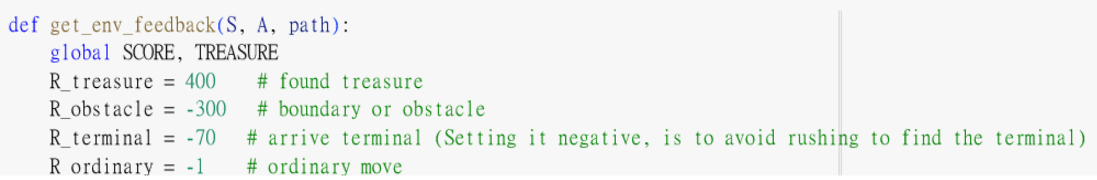
四、 參數設定

- 建置迷宮的參數
- RL 的參數(EPSILON、ALPHA、GAMMA)：發現貪婪指數調高、learning rate 調低、專注長期效益會有助於找寶藏。

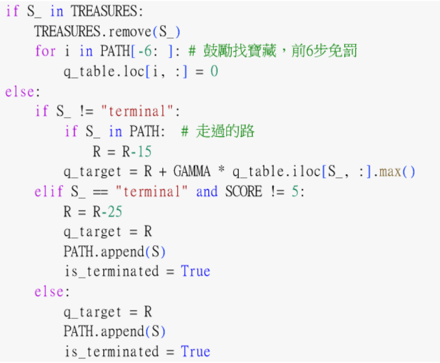


```
N_STATES_x = 21
N_STATES_y = 11
ACTIONS = ["left", "right", "up", "down"]
GOAL = 230
EPSILON = 0.95    # greedy
ALPHA = 0.05      # learning-rate
GAMMA = 0.95      # focus on long-term learning
MAX_EPISODES = 800
FRESH_TIME = 0
```

- Reward



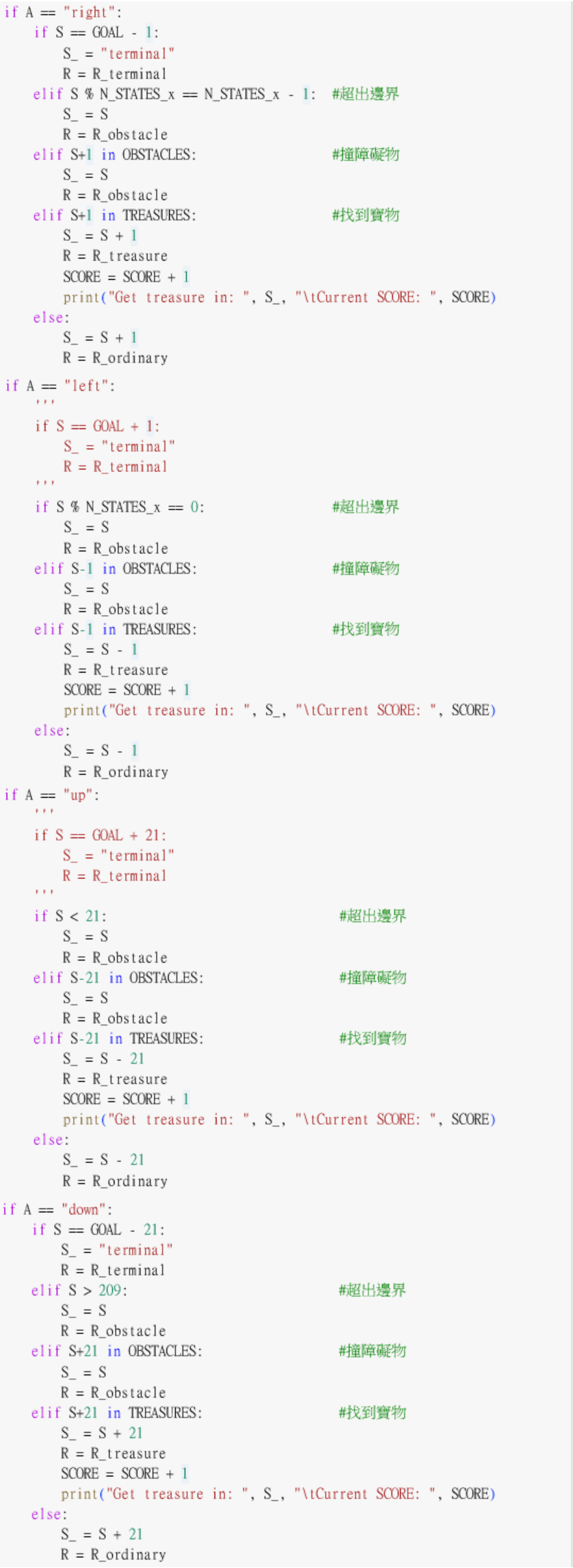
```
def get_env_feedback(S, A, path):
    global SCORE, TREASURE
    R_treasure = 400    # found treasure
    R_obstacle = -300   # boundary or obstacle
    R_terminal = -70   # arrive terminal (Setting it negative, is to avoid rushing to find the terminal)
    R_ordinary = -1    # ordinary move
```



```
if S_ in TREASURES:
    TREASURES.remove(S_)
    for i in PATH[-6: ]: # 鼓勵找寶藏，前6步免罰
        q_table.loc[i, :] = 0
else:
    if S_ != "terminal":
        if S_ in PATH:  # 走過的路
            R = R-15
        q_target = R + GAMMA * q_table.iloc[S_, :].max()
    elif S_ == "terminal" and SCORE != 5:
        R = R-25
        q_target = R
        PATH.append(S)
        is_terminated = True
    else:
        q_target = R
        PATH.append(S)
        is_terminated = True
```

- R_treasure：400	找到寶藏 (鼓勵找寶藏， 值為正數)
- R_obstacle：-300	撞牆與超出邊界 (跟找到寶藏相對， 值為負數)
- R_terminal：-70	抵達終點 (之所以為負數是因為發現設為正數的話， 會直衝終點。為避免直衝終點的現象， 故設為負數)
- R_ordinary：-1	一般移動
- 走過的路：-15	限制不要往回走，故為負值比一般移動扣更多
- 沒找完寶藏：-25	鼓勵找寶藏
- 找寶藏免罰	鼓勵找寶藏

- Move (Up、Right、Down、Left)



```
if A == "right":
    if S == GOAL - 1:
        S_ = "terminal"
        R = R_terminal
    elif S % N_STATES_x == N_STATES_x - 1:  #超出邊界
        S_ = S
        R = R_obstacle
    elif S+1 in OBSTACLES:                  #撞障礙物
        S_ = S
        R = R_obstacle
    elif S+1 in TREASURES:                  #找到寶物
        S_ = S + 1
        R = R_treasure
        SCORE = SCORE + 1
        print("Get treasure in: ", S_, "\tCurrent SCORE: ", SCORE)
    else:
        S_ = S + 1
        R = R_ordinary
if A == "left":
    '''
    if S == GOAL + 1:
        S_ = "terminal"
        R = R_terminal
    '''
    if S % N_STATES_x == 0:                 #超出邊界
        S_ = S
        R = R_obstacle
    elif S-1 in OBSTACLES:                  #撞障礙物
        S_ = S
        R = R_obstacle
    elif S-1 in TREASURES:                  #找到寶物
        S_ = S - 1
        R = R_treasure
        SCORE = SCORE + 1
        print("Get treasure in: ", S_, "\tCurrent SCORE: ", SCORE)
    else:
        S_ = S - 1
        R = R_ordinary
if A == "up":
    '''
    if S == GOAL + 21:
        S_ = "terminal"
        R = R_terminal
    '''
    if S < 21:                               #超出邊界
        S_ = S
        R = R_obstacle
    elif S-21 in OBSTACLES:                  #撞障礙物
        S_ = S
        R = R_obstacle
    elif S-21 in TREASURES:                  #找到寶物
        S_ = S - 21
        R = R_treasure
        SCORE = SCORE + 1
        print("Get treasure in: ", S_, "\tCurrent SCORE: ", SCORE)
    else:
        S_ = S - 21
        R = R_ordinary
if A == "down":
    if S == GOAL - 21:
        S_ = "terminal"
        R = R_terminal
    elif S > 209:                            #超出邊界
        S_ = S
        R = R_obstacle
    elif S+21 in OBSTACLES:                  #撞障礙物
        S_ = S
        R = R_obstacle
    elif S+21 in TREASURES:                  #找到寶物
        S_ = S + 21
        R = R_treasure
        SCORE = SCORE + 1
        print("Get treasure in: ", S_, "\tCurrent SCORE: ", SCORE)
    else:
        S_ = S + 21
        R = R_ordinary
```

五、　心得

雖然這次作業的程式碼看似是三個作業中最不複雜的一個，但實際上需要花不少時間去理解運作的機制。原因就在於有太多的參數以及訓練模式的可能性，需要慢慢地去嘗試跟調整。

最初，我的訓練狀況呈現一個很極端的慘狀，不是因為找寶藏 Episode 步數極大，就是後面幾次 Episode 都直衝終點，可能是找寶藏的代價太高了。經過多次參數的嘗試，並加上「未找滿寶藏下，抵達終點則判罰」的條件，步數能壓在 400 步左右。接著，在 400 步左右，便是大卡關，一直都壓不到 300 步以下。幸運的是，後來透過身邊朋友的提點，說可以嘗試「若找到寶藏，則前幾步不判罰」的條件，必須要說，這想法實在是太厲害了，一舉把訓練壓到 300 步內。(感謝好友~)

整體來說，這份作業是份很有趣的作業，除了調整(也可以說是在玩)參數，跟其他人想法上的交流，因而迸出新火花，都是很寶貴的學習經驗。

六、　參考資料

https://www.samyzaf.com/ML/rl/qmaze.html

https://medium.com/data-science-in-your-pocket/maze-runner-%EF%B8%8F-with-off-policy-q-learning-no-back-stepping-allowed-d01a79a6199c