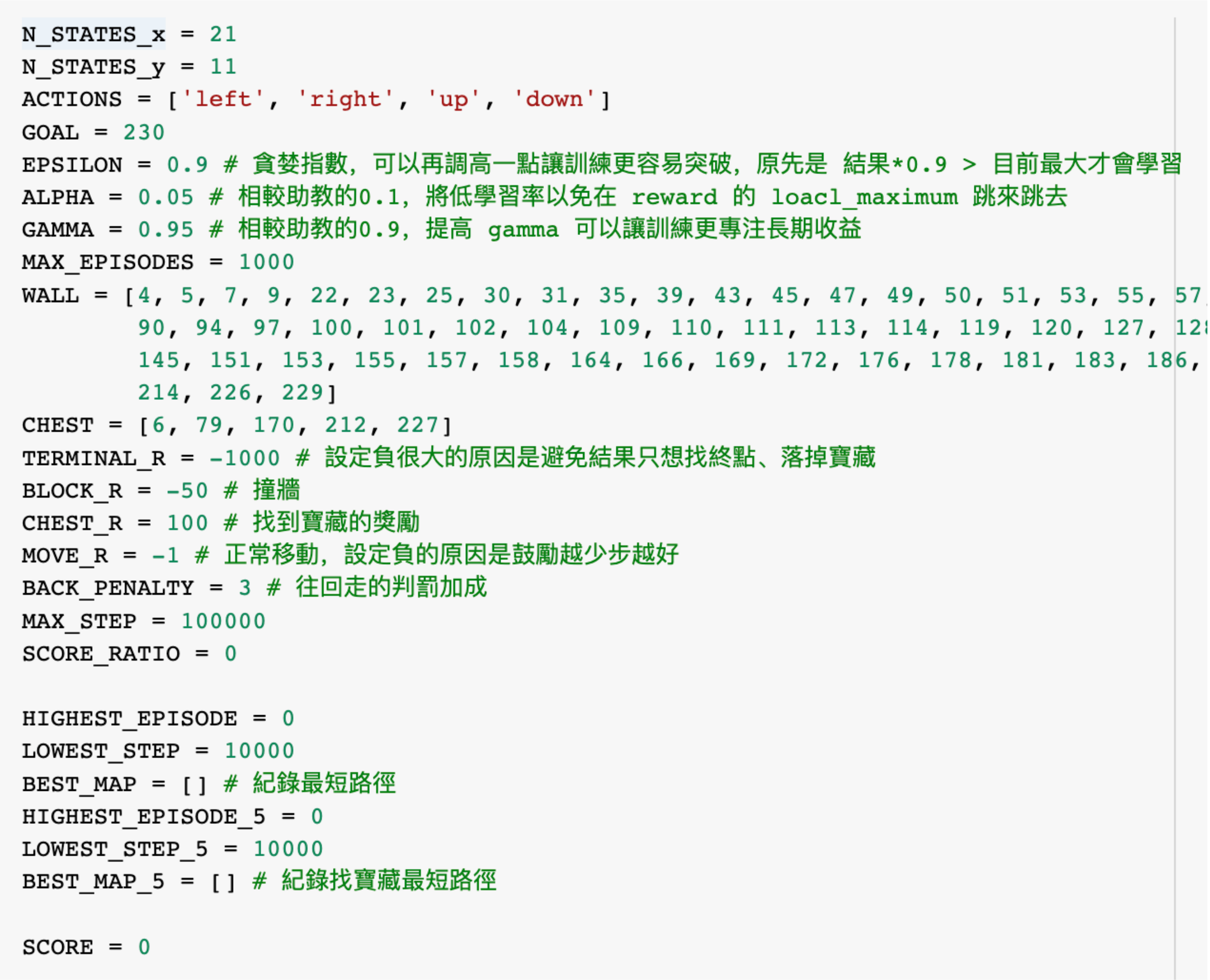
AI 第三次作業 Maze

資管二A 109403524 洪祥銘

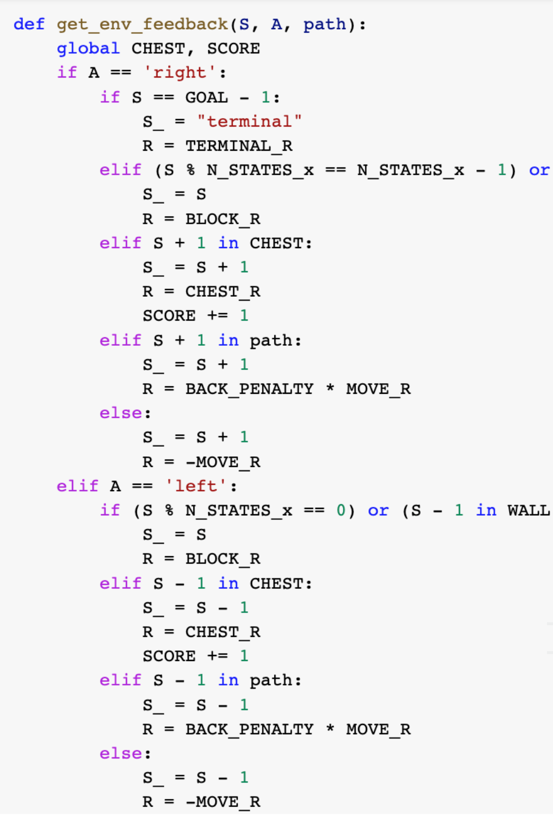
Colab Link : https://colab.research.google.com/drive/12LTis5s4UZeBJ23xcjxHXteozrPTdhsP?usp=sharing

參數設定



```
N_STATES_x = 21
N_STATES_y = 11
ACTIONS = ['left', 'right', 'up', 'down']
GOAL = 230
EPSILON = 0.9 # 貪婪指數，可以再調高一點讓訓練更容易突破，原先是 結果*0.9 > 目前最大才會學習
ALPHA = 0.05 # 相較助教的0.1，將低學習率以免在 reward 的 loacl_maximum 跳來跳去
GAMMA = 0.95 # 相較助教的0.9，提高 gamma 可以讓訓練更專注長期收益
MAX_EPISODES = 1000
WALL = [4, 5, 7, 9, 22, 23, 25, 30, 31, 35, 39, 43, 45, 47, 49, 50, 51, 53, 55, 57
        90, 94, 97, 100, 101, 102, 104, 109, 110, 111, 113, 114, 119, 120, 127, 12
        145, 151, 153, 155, 157, 158, 164, 166, 169, 172, 176, 178, 181, 183, 186,
        214, 226, 229]
CHEST = [6, 79, 170, 212, 227]
TERMINAL_R = -1000 # 設定負很大的原因是避免結果只想找終點、落掉寶藏
BLOCK_R = -50 # 撞牆
CHEST_R = 100 # 找到寶藏的獎勵
MOVE_R = -1 # 正常移動，設定負的原因是鼓勵越少步越好
BACK_PENALTY = 3 # 往回走的判罰加成
MAX_STEP = 100000
SCORE_RATIO = 0

HIGHEST_EPISODE = 0
LOWEST_STEP = 10000
BEST_MAP = [] # 紀錄最短路徑
HIGHEST_EPISODE_5 = 0
LOWEST_STEP_5 = 10000
BEST_MAP_5 = [] # 紀錄找寶藏最短路徑

SCORE = 0
```

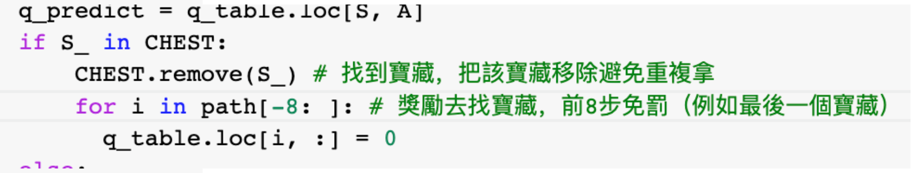
Reward 設定



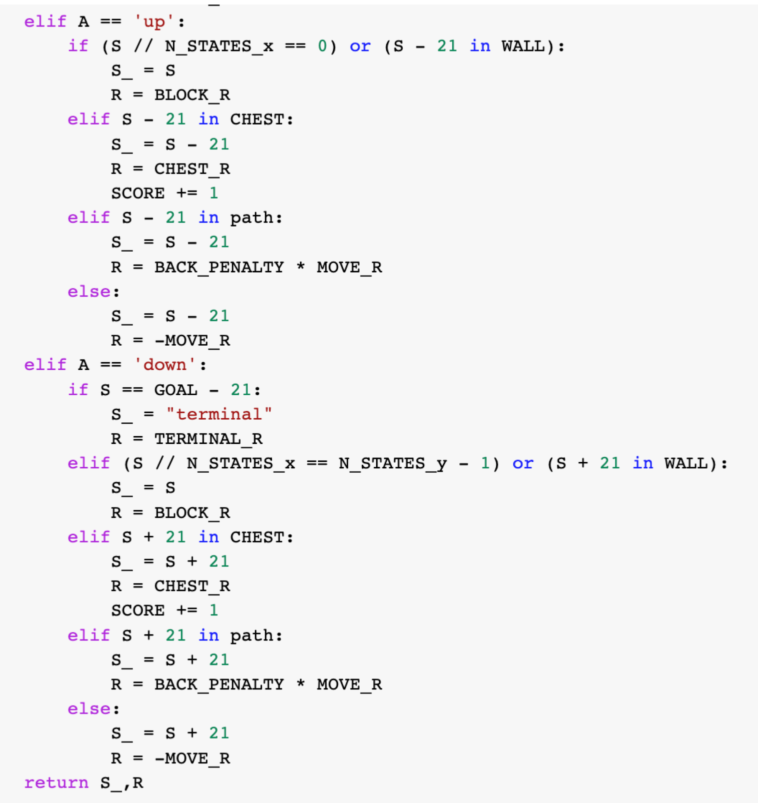
```
def get_env_feedback(S, A, path):
    global CHEST, SCORE
    if A == 'right':
        if S == GOAL - 1:
            S_ = "terminal"
            R = TERMINAL_R
        elif (S % N_STATES_x == N_STATES_x - 1) or (S + 1 in WALL):
            S_ = S
            R = BLOCK_R
        elif S + 1 in CHEST:
            S_ = S + 1
            R = CHEST_R
            SCORE += 1
        elif S + 1 in path:
            S_ = S + 1
            R = BACK_PENALTY * MOVE_R
        else:
            S_ = S + 1
            R = -MOVE_R
    elif A == 'left':
        if (S % N_STATES_x == 0) or (S - 1 in WALL):
            S_ = S
            R = BLOCK_R
        elif S - 1 in CHEST:
            S_ = S - 1
            R = CHEST_R
            SCORE += 1
        elif S - 1 in path:
            S_ = S - 1
            R = BACK_PENALTY * MOVE_R
        else:
            S_ = S - 1
            R = -MOVE_R
```

以每個移動方向內的 if - block 做區分

1. 如果遇到終點 reward – 1000
   目的是鼓勵去找寶藏，避免訓練出只找終點的結果
2. 撞牆 reward -50
3. 找到寶藏 reward +100，外加下圖的往回走減免
4. 往回走會有三倍的額外懲罰，見 5.
5. 如果只是普通往前 reward -1



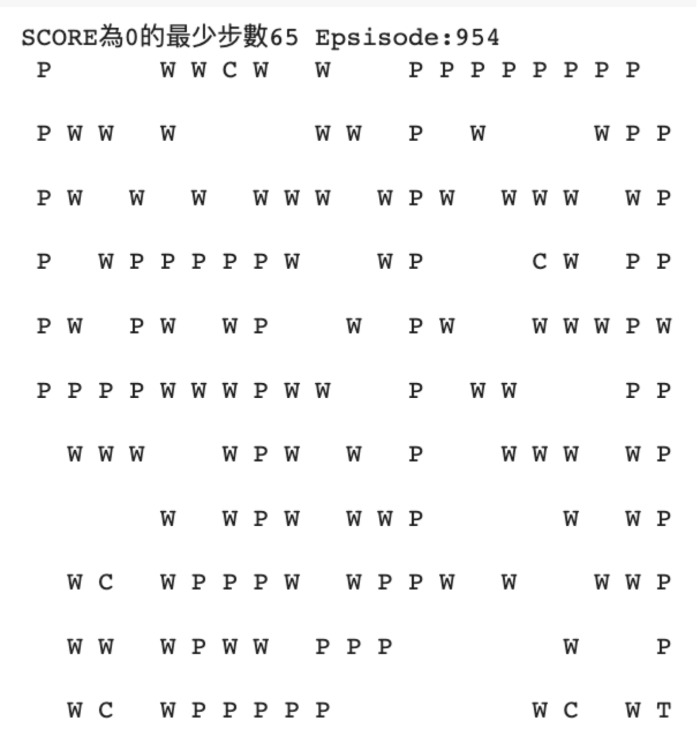
```
q_predict = q_table.loc[S, A]
if S_ in CHEST:
    CHEST.remove(S_) # 找到寶藏，把該寶藏移除避免重複拿
    for i in path[-8: ]: # 獎勵去找寶藏，前8步免罰（例如最後一個寶藏）
      q_table.loc[i, :] = 0
else:
```



```
    elif A == 'up':
        if (S // N_STATES_x == 0) or (S - 21 in WALL):
            S_ = S
            R = BLOCK_R
        elif S - 21 in CHEST:
            S_ = S - 21
            R = CHEST_R
            SCORE += 1
        elif S - 21 in path:
            S_ = S - 21
            R = BACK_PENALTY * MOVE_R
        else:
            S_ = S - 21
            R = -MOVE_R
    elif A == 'down':
        if S == GOAL - 21:
            S_ = "terminal"
            R = TERMINAL_R
        elif (S // N_STATES_x == N_STATES_y - 1) or (S + 21 in WALL):
            S_ = S
            R = BLOCK_R
        elif S + 21 in CHEST:
            S_ = S + 21
            R = CHEST_R
            SCORE += 1
        elif S + 21 in path:
            S_ = S + 21
            R = BACK_PENALTY * MOVE_R
        else:
            S_ = S + 21
            R = -MOVE_R
    return S_,R
```

到達終點（0 個寶藏最少步數）

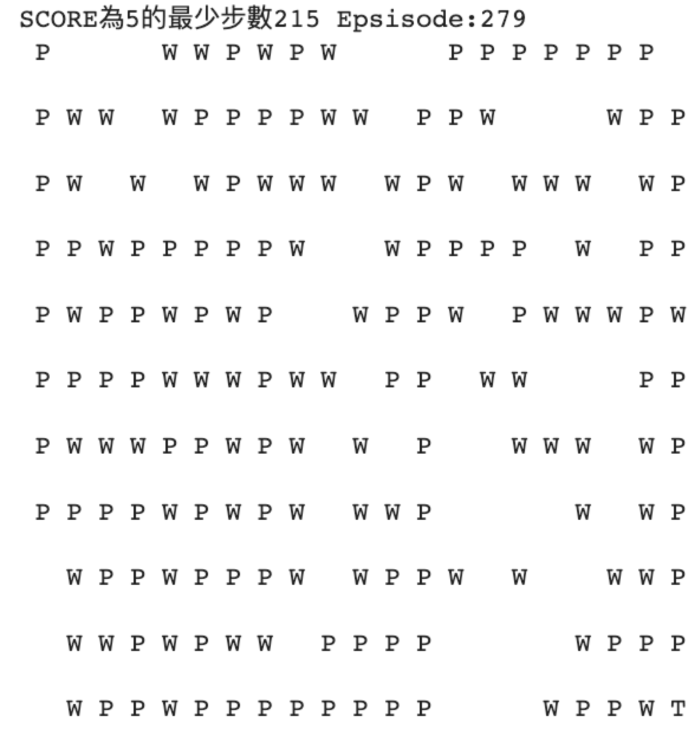
最少 65 步（理論最佳可以 61）



```
SCORE為0的最少步數65 Epsisode:954
 P       W W C W   W     P P P P P P P P

 P W W   W         W W   P   W       W P P

 P W   W   W   W W W   W P W   W W W   W P

 P   W P P P P P W     W P       C W   P P

 P W   P W   W P     W   P W     W W W P W

 P P P P W W W P W W     P   W W       P P

   W W W     W P W   W   P     W W W   W P

         W   W P W   W W P         W   W P

   W C   W P P P W   W P P W   W     W W P

   W W   W P W W   P P P           W     P

   W C   W P P P P P             W C   W T
```

寶藏全拿（5 個寶藏的最少步數）

最少 215 步，有同學做出 194，但他的 case 的 0 個寶藏狀況普普，權衡之下這個結果應該不錯



```
SCORE為5的最少步數215 Epsisode:279
 P       W W P W P W       P P P P P P P

 P W W   W P P P P W W   P P W       W P P

 P W   W   W P W W W   W P W   W W W   W P

 P P W P P P P P W     W P P P P   W   P P

 P W P P W P W P     W P P W   P W W W P W

 P P P P W W W P W W   P P   W W       P P

 P W W W P P W P W   W   P     W W W   W P

 P P P P W P W P W   W W P         W   W P

   W P P W P P P W   W P P W   W     W W P

   W W P W P W W   P P P P         W P P P

   W P P W P P P P P P P P       W P P W T
```

心得

首先助教辛苦了，改一學期的作業。
沒意外的話這是這學期的最後一份作業，我三份估計做的都還不錯，第一份 CNN 的準確率 70 up，RNN 的應該大家都差不多，這一份作業看起來也蠻鬼的，目前我們一起研究作業的人沒看到可以做到 65 / 215 的情況，這三份基本上都有先人的智慧加上我後天得運用得當，運氣很好、讀到的文章都有用 CSDN、StackOverflow 大法好。這三份作業做下來有個大心得就是基本上大方向把握好都可以滿分，但要去 tweak 結果其實只要有個 mindset『改學習率、套 optimizer』去優化效果比較好，理論上去對資料集、input 值去做大改沒什麼用，數學問題還是要數學解決；另外，看文件的能力也有顯著提升，除了 TF、Keras 本身的文檔，也蠻多部落客文章有分享一些比較，蠻多結果可以應用在作業上的。