此次作業是用 Q-Learning 進行尋找寶藏，範例程式是利用一維的地圖，動作也只會有左和右兩種，作業的要求是要走 30 x 40 大小的地圖，所以要先做的兩件事分別為：一、將地圖從一維改成二維；二、將動作由兩種改為四種，原本左和右兩種，之後再加上上跟下兩種。

```
x_len=40  #y軸
y_len=30  #x軸
SPACE = np.zeros((x_len,y_len))
N_STATES = [ ss for ss in range(SPACE.size)]
N_STATES_dict = {}
```

儲存空間改成二維

```
ACTIONS = ['left', 'right', 'up', 'down']     #可以做的動作
```

```
GOAL = [39,29]      #寶藏位置
```

寶藏設在右下角

再來要更動的部分是在於環境對我們行為的 feedback，原本只要考慮左跟右，現在要再加上上跟下，然後會多一個 current_location 的變數來檢查是否有超過設定的邊界範圍，最後如果抵達寶藏地之後，會給予 100 的 Reward

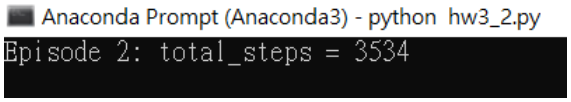
```
#--------建立環境對我們行為的feedback---------
def get_env_feedback(S, A):
  current_location = N_STATES_dict[S].copy()
  if (A == 'right') and (current_location[0]<(x_len-1)):
    current_location[0] += 1
  if (A == 'left') and (current_location[0]>0):
    current_location[0] -= 1
  if (A == 'up') and (current_location[1]>0):
    current_location[1] -= 1
  if (A == 'down') and (current_location[1]<(y_len-1)):
    current_location[1] += 1

  if current_location!=GOAL:
    R = 0 #reward
    S_ = get_key(N_STATES_dict,current_location)[-1]
  if current_location==GOAL:
    R = 100
    S_ = 'terminal'
  return S_ ,R
```

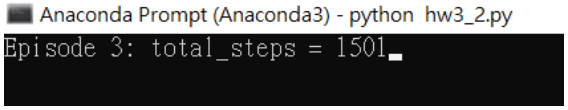
因為有考量到地圖比起之前一維的範例要大上很多，要找出最佳路徑的回合數恐怕訓練次數需要要增加不少，故先將 Episode 的次數改為 50 次。

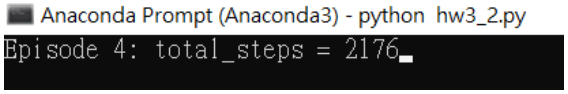
```
MAX_EPISODES = 50   # maximum episodes
```

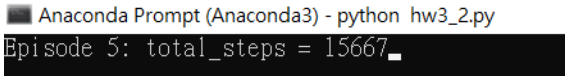
上述為主要的程式碼改動，接著就是用上述改過後的程式碼去進行遊戲。

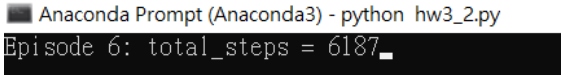
```
Anaconda Prompt (Anaconda3) - python  hw3_2.py
Episode 1: total_steps = 46075
```

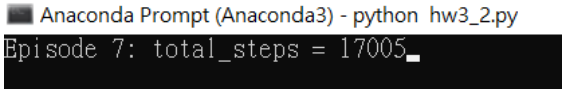


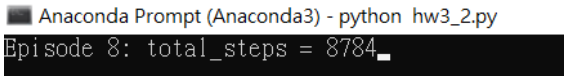


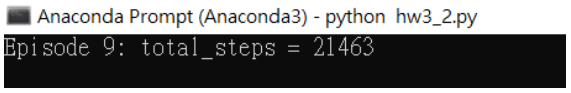


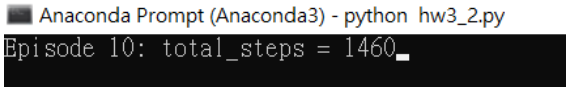




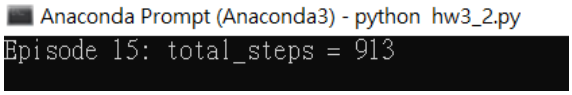






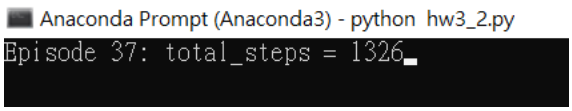


前十次的結果大多數都不太好，有四次的總和數超過一萬，最好的一次也需要經過 1460 步；原本想說前面十次之後步數應該會越來越少，但是在在第二十至三十的 Episode 之間還是有多次需要破萬步才能找到寶藏的所在地，不過有一次只花 913 步。

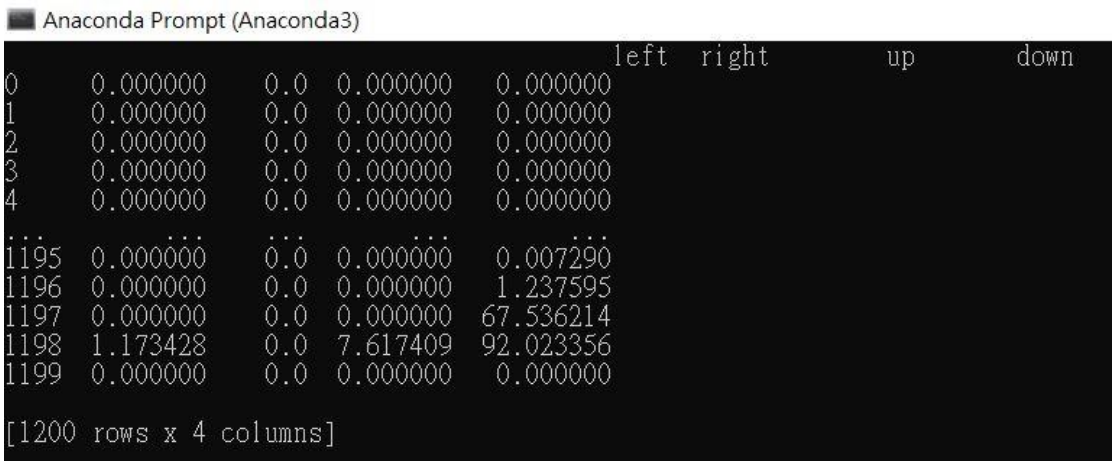


在第二十至三十的 Episode 之間，總回合數都已降至一萬步以下，雖然還是會有六、七千步的結果出現，但也有出現一次只需 437 步的結果。

在第三十至四十的 Episode 之間，總回合數都在數千步，最好的一次為第 37 個 Episode 的 1326 步，這十次並沒有較好的結果。



在第四十至五十的 Episode 之間，總回合數有三次來到一千步以內，但並未比 437 次低，故最低的回合數為 437 次，下方為最後的 q table。



最好的結果：437steps

這次的作業其實也是需要耗費一定時間，因為每經過一次 episode 都要花不少時間，一個多小時可能才跑了不到 10 個 episode，跑了兩次就花費了整整一天的時間；自己覺得此次作業，在找最好的回合數上應該再多花些功夫，我覺得還有很大的進步空間。