作業 3 – Maze

資管三_109403502_楊珮綾

- **Colab 連結:**
  https://colab.research.google.com/drive/1OFeq269I8VuM0t3OlbCOv71BK4Bz8RaL?usp=sharing

- **訓練完的 Q-table:**

Q-table:

	left	right	up	down
0	-14.705308	-9.402874	-14.008818	-9.405743
1	-9.405475	-9.408735	-15.788222	-14.140432
2	-9.411287	-9.411250	-15.174791	-15.169201
3	-9.412735	-12.177549	-15.765102	-9.417713
4	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000
6	-17.733525	-17.765246	-17.583425	-8.911326
7	0.000000	0.000000	0.000000	0.000000
8	-11.820561	-14.163369	-11.597459	-9.128415
9	0.000000	0.000000	0.000000	0.000000
10	-7.602411	-7.440988	-7.725199	-7.701580
11	-7.436202	-7.438109	-11.780378	-7.424841
12	-7.397600	-7.408234	-12.248502	-7.406897
13	-7.279773	-7.243332	-11.751021	-7.249381
14	-7.027228	-6.965031	-7.581345	-10.188359
15	-6.716834	-6.544053	-10.395295	-6.690666
16	-6.577063	-5.883971	-10.820649	-6.408441
17	-6.108961	-4.826904	-10.554009	-6.195382
18	-5.722003	-3.077752	-8.142193	-10.861670
19	-5.457730	-5.265835	-6.521090	-0.239477
20	-4.965207	-5.771382	-7.127007	-4.722013
21	-14.960286	-15.704267	-9.386229	-9.386175
22	0.000000	0.000000	0.000000	0.000000
23	0.000000	0.000000	0.000000	0.000000
24	-13.419532	-12.979532	-9.417666	-11.009108
25	0.000000	0.000000	0.000000	0.000000
26	-10.921024	-9.137869	-10.932521	-13.227409
27	-9.144765	-9.152909	-8.898880	-9.146046
28	-9.092098	-9.125851	-9.765940	-11.967422
29	-9.129555	-12.176744	-9.127323	-12.235418
30	0.000000	0.000000	0.000000	0.000000
31	0.000000	0.000000	0.000000	0.000000
32	-9.346071	-7.482404	-7.460506	-8.441424
33	-7.509195	-7.512664	-7.510496	-7.558713
34	-7.394606	-11.768533	-7.391865	-9.711974
35	0.000000	0.000000	0.000000	0.000000
36	-8.181597	-6.573773	-6.579228	-9.125806
37	-6.446517	-6.439242	-6.425447	-8.660405
38	-6.298057	-8.281683	-6.226692	-7.615383
39	0.000000	0.000000	0.000000	0.000000
40	-7.208026	4.068228	-5.031922	-7.014009
41	-4.027178	-6.951434	-4.677307	9.520803
42	-14.534576	-14.975782	-9.365382	-9.368897
43	0.000000	0.000000	0.000000	0.000000
44	0.000000	0.000000	0.000000	0.000000
45	0.000000	0.000000	0.000000	0.000000
46	-11.504612	-10.129569	-9.702641	-9.182293
47	0.000000	0.000000	0.000000	0.000000
48	-12.897909	-14.735429	-9.115412	-9.118914
49	0.000000	0.000000	0.000000	0.000000
50	0.000000	0.000000	0.000000	0.000000
51	0.000000	0.000000	0.000000	0.000000
52	-9.167562	-10.115285	-10.853849	-9.050871
53	0.000000	0.000000	0.000000	0.000000
54	-10.620828	-9.662626	-7.677858	-7.704970
55	0.000000	0.000000	0.000000	0.000000
56	-8.622400	-10.685065	-9.201091	-7.826752
57	0.000000	0.000000	0.000000	0.000000
58	0.000000	0.000000	0.000000	0.000000
59	0.000000	0.000000	0.000000	0.000000
60	-3.940399	-4.734508	-3.940399	-3.070694
61	0.000000	0.000000	0.000000	0.000000
62	-4.714682	-4.626758	-4.051566	16.611394
63	-14.958642	-9.343700	-9.350038	-9.347997
64	-9.344023	-12.306966	-13.875896	-11.081061
65	0.000000	0.000000	0.000000	0.000000
66	-12.118768	-9.206783	-15.062314	-9.209468
67	-9.179997	-9.179453	-9.180896	-14.839140
68	-9.151730	-9.149191	-15.224008	-9.150197
69	-9.120626	-9.117257	-9.115414	-13.362564
70	-9.084520	-13.700046	-12.688780	-9.081933
71	0.000000	0.000000	0.000000	0.000000
72	-14.092522	-9.046433	-13.303458	-9.050157
73	-9.051188	-12.611259	-9.049045	-11.490666
74	0.000000	0.000000	0.000000	0.000000
75	-11.458273	-7.826935	-7.824720	-7.829342
76	-7.834189	-7.825049	-10.514272	-8.476596
77	-7.838568	-7.833548	-7.825104	-7.835172
78	-7.856155	-7.510293	-15.483631	-7.830919
79	-7.634118	-14.239105	-15.401678	-15.167473
80	0.000000	0.000000	0.000000	0.000000
81	-3.930258	-2.782677	-3.068313	-4.713834
82	-3.132300	-2.957764	-4.748205	34.073340
83	24.805153	-5.812640	-1.971837	-4.860782
84	-15.186469	-13.624390	-9.325428	-9.329669
85	0.000000	0.000000	0.000000	0.000000
86	-15.488895	-9.247835	-14.196677	-9.248723
87	-9.236154	-16.005220	-9.235806	-9.232580
88	0.000000	0.000000	0.000000	0.000000
89	-11.449614	-11.441031	-9.151309	-9.325473
90	0.000000	0.000000	0.000000	0.000000
91	-15.150689	-9.046075	-9.048220	-9.047159
92	-9.049893	-9.045411	-12.334379	-14.213118
93	-9.050503	-13.467929	-9.044606	-14.401320
94	0.000000	0.000000	0.000000	0.000000
95	-8.563444	-8.003380	-10.647432	-8.000046
96	-7.947472	-12.350197	-7.947836	-7.972191
97	0.000000	0.000000	0.000000	0.000000
98	-11.041843	-7.818627	-7.834976	-9.892026
99	-7.816683	-8.409078	-7.793856	-8.563834
100	0.000000	0.000000	0.000000	0.000000
101	0.000000	0.000000	0.000000	0.000000
102	0.000000	0.000000	0.000000	0.000000
103	0.216309	4.674270	2.759922	42.972721
104	0.000000	0.000000	0.000000	0.000000
105	-13.288903	-9.306692	-9.308092	-9.305215
106	-9.282169	-9.286052	-14.917046	-15.150697
107	-9.263664	-9.263019	-9.261189	-16.742658
108	-9.250637	-14.506662	-9.248166	-13.123996
109	0.000000	0.000000	0.000000	0.000000
110	0.000000	0.000000	0.000000	0.000000
111	0.000000	0.000000	0.000000	0.000000
112	-14.792458	-11.957443	-9.011370	-9.007310
113	0.000000	0.000000	0.000000	0.000000
114	0.000000	0.000000	0.000000	0.000000
115	-9.485054	-8.049505	-10.083302	-8.423631
116	-8.046505	-8.041636	-8.049532	-8.052628
117	-8.051784	-8.039888	-8.042529	-8.054820
118	-8.084364	-10.032246	-10.468908	-8.071074
119	0.000000	0.000000	0.000000	0.000000
120	0.000000	0.000000	0.000000	0.000000
121	-2.939744	-1.980640	-1.990000	-2.893166
122	-1.977446	-1.977884	-1.990000	-2.849013
123	-1.972829	2.059159	-3.715075	-1.977371
124	-2.026019	52.179000	3.508152	0.795138
125	9.360890	-2.901203	6.952762	60.822973
126	-13.482611	-13.622432	-9.307229	-9.304849
127	0.000000	0.000000	0.000000	0.000000
128	0.000000	0.000000	0.000000	0.000000
129	0.000000	0.000000	0.000000	0.000000
130	-9.337616	-8.786401	-10.082257	-9.375897
131	-8.782129	-10.748125	-11.558409	-8.791848
132	0.000000	0.000000	0.000000	0.000000
133	-12.973885	-14.898971	-8.966635	-8.968366
134	0.000000	0.000000	0.000000	0.000000
135	-8.576456	-10.766414	-8.585225	-8.392109
136	0.000000	0.000000	0.000000	0.000000
137	-10.088137	-8.083497	-8.087455	-10.786415
138	-8.119666	-8.105691	-8.103045	-8.089416
139	-8.121718	-8.105951	-8.121078	-8.122508
140	-8.151869	-11.052476	-11.221781	-8.147715
141	0.000000	0.000000	0.000000	0.000000
142	0.000000	0.000000	0.000000	0.000000
143	0.000000	0.000000	0.000000	0.000000
144	-1.990000	-3.720892	-1.988569	-1.987351
145	0.000000	0.000000	0.000000	0.000000
146	-1.000000	3.374028	-0.767055	69.708206
147	-15.382415	-9.304710	-9.307925	-9.306172
148	-9.307043	-9.304015	-15.167992	-13.799298
149	-9.329562	-9.319343	-17.943972	-8.775184
150	-9.263062	-15.591112	-14.775010	-9.306126
151	0.000000	0.000000	0.000000	0.000000
152	-10.667116	-10.514355	-8.783905	-8.789269
153	0.000000	0.000000	0.000000	0.000000
154	-14.255668	-13.511087	-8.926940	-8.925475
155	0.000000	0.000000	0.000000	0.000000
156	-10.505156	-10.730099	-8.391352	-8.393252
157	0.000000	0.000000	0.000000	0.000000
158	0.000000	0.000000	0.000000	0.000000
159	-11.358170	-8.153918	-8.165247	-8.171533
160	-8.142167	-8.177319	-8.141411	-12.380979
161	-8.182498	-8.179624	-8.185628	-8.178818
162	-8.191756	-8.190630	-12.224616	-10.579870
163	-8.205336	-10.767309	-12.694769	-8.195561
164	0.000000	0.000000	0.000000	0.000000
165	-1.990000	-2.962853	-1.990239	-1.953797
166	0.000000	0.000000	0.000000	0.000000
167	9.601218	5.199320	4.829444	78.990447
168	-12.567503	-13.841390	-9.307566	-9.305267
169	0.000000	0.000000	0.000000	0.000000
170	-18.011052	-9.330275	-8.945785	-18.068341
171	-9.293250	-16.628307	-9.308487	-9.307119
172	0.000000	0.000000	0.000000	0.000000
173	-15.674715	-8.793779	-8.785339	-8.784204
174	-8.835553	-8.848751	-13.488033	-13.940104
175	-8.885186	-12.293451	-8.882191	-11.973070
176	0.000000	0.000000	0.000000	0.000000
177	-10.948953	-11.260228	-8.389536	-8.402729
178	0.000000	0.000000	0.000000	0.000000
179	-13.136190	-8.279033	-12.509730	-8.274535
180	-8.239996	-11.375557	-8.241221	-8.245580
181	0.000000	0.000000	0.000000	0.000000
182	-10.677155	-9.035234	-8.208471	-8.195900
183	0.000000	0.000000	0.000000	0.000000
184	-12.751677	-8.204305	-8.212130	-8.211779
185	-8.206618	-9.042628	-9.178263	-8.429867
186	0.000000	0.000000	0.000000	0.000000
187	0.000000	0.000000	0.000000	0.000000
188	11.869896	6.028467	22.337027	88.978527
189	-12.643977	-13.455273	-9.308554	-9.304491
190	0.000000	0.000000	0.000000	0.000000
191	0.000000	0.000000	0.000000	0.000000
192	-12.472194	-14.904683	-9.305980	-9.298508
193	0.000000	0.000000	0.000000	0.000000
194	-13.225251	-11.014716	-8.735843	-8.726654
195	0.000000	0.000000	0.000000	0.000000
196	0.000000	0.000000	0.000000	0.000000
197	-11.337166	-8.437759	-12.413931	-8.438907
198	-8.403169	-8.405983	-8.401483	-8.412815
199	-8.369866	-8.368377	-13.379457	-8.362525
200	-8.320299	-8.301741	-8.315131	-8.308061
201	-8.269759	-8.267603	-8.279184	-8.270652
202	-8.255907	-8.241913	-10.181318	-8.242111

203	-8.220252	-8.221462	-8.230657	-8.233596
204	-8.217371	-8.215998	-12.349965	-8.229166
205	-8.207902	-10.528605	-8.221208	-10.550539
206	0.000000	0.000000	0.000000	0.000000
207	-1.000000	-0.199000	0.000000	0.000000
208	-0.190000	8.809734	-1.000000	-1.000000
209	-0.190000	-1.000000	32.683332	99.997344
210	-11.274617	-11.551950	-9.306956	-13.357398
211	0.000000	0.000000	0.000000	0.000000
212	-17.986770	-9.331400	-17.972216	-18.006176
213	-9.328801	-17.804016	-9.328059	-18.050920
214	0.000000	0.000000	0.000000	0.000000
215	-11.700719	-8.669155	-8.669277	-11.260382
216	-8.609813	-8.607925	-10.970670	-10.674002
217	-8.544708	-8.545792	-10.862421	-11.840482
218	-8.485765	-8.476416	-8.471629	-14.830598
219	-8.427545	-8.421908	-8.414003	-12.330240
220	-8.363678	-8.373154	-8.377721	-12.788060
221	-8.335958	-8.317630	-8.320763	-11.061400
222	-8.278342	-8.281631	-8.285296	-10.153297
223	-8.256445	-8.260497	-8.248612	-11.517222
224	-8.239971	-8.237142	-8.251269	-12.936870
225	-8.234490	-10.086697	-8.232331	-10.546863
226	0.000000	0.000000	0.000000	0.000000
227	0.000000	0.000000	0.000000	0.000000
228	0.000000	0.000000	0.000000	0.000000
229	0.000000	0.000000	0.000000	0.000000
230	0.000000	0.000000	0.000000	0.000000

- **步數最少且寶藏數最多的截圖（步數 +score 分數）**

總步數: 377

SCORE: 3

- **Reward 設定截圖並說明**

右:

情況 1：當下位置已經在右邊邊界，S 不走動(S_ = S)，且 R =-10

情況 2：當下位置在牆的左邊，S 不走動(S_ = S)，且 R =-10

情況 3：當下位置已經在終點，最後位置就是 terminal，R=100

情況 4：往右走一步的位置是有寶藏的，S 可以往右(S_=S+1)，且 R=10，把得過得寶藏從寶藏陣列去除，並且得 1 分

情況 5：上述情況外，直接右走(S_=S+1)，R=-1

```
def get_env_feedback(S, A, path):
        global SCORE
        global TREASURES
        global WALLS
        if A == "right":
                if S % N_STATES_x == N_STATES_x - 1:    # 到達右邊界
                        S_ = S
                        R = -10
                elif (S+1) in WALLS:    # 撞到牆壁
                        S_ = S
                        R = -10
                elif S == GOAL:
                        S_ = "terminal"
                        R = 100
                elif (S +1) in TREASURES:
                        S_ = S +1
                        R = 10
                        TREASURES.remove(S_)
                        SCORE=SCORE+1
                else:
                        S_ = S + 1    # 向右移動
                        R = -1
```

左:
情況 1：當下位置已經在左邊邊界，S 不走動(S_ = S)，且 R =-10
情況 2：當下位置在牆的右邊，S 不走動(S_ = S)，且 R =-10
情況 3：當下位置已經在終點，最後位置就是 terminal，R=100
情況 4：往左走一步的位置是有寶藏的，S 可以往左(S_=S-1)，且 R=10，把得過得寶藏從寶藏陣列去除，並且得 1 分
情況 5：上述情況外，直接左走(S_=S-1)，R=-1

```
elif A == "left":
        if S % N_STATES_x == 0 :    # 到達左邊界
                S_ = S
                R = -10
        elif (S-1) in WALLS:  # 撞到牆壁
                S_ = S
                R = -10
        elif S == GOAL:
                S_ = "terminal"
                R = 100
        elif (S -1) in TREASURES:
                S_ = S -1
                R = 10
                TREASURES.remove(S_)
                SCORE=SCORE+1
        else:
                S_ = S - 1    # 向左移動
                R = -1
```

上:
情況 1：當下位置已經在上面邊界，S 不走動(S_ = S)，且 R =-10
情況 2：當下位置在牆的下面，S 不走動(S_ = S)，且 R =-10
情況 3：當下位置已經在終點，最後位置就是 terminal，R=100
情況 4：往上走一步的位置是有寶藏的，S 可以往上(S_=S-N_STATES_x)，且 R=10，把得過得寶藏從寶藏陣列去除，並且得 1 分
情況 5：上述情況外，直接上走(S_= S-N_STATES_x)，R=-1

```
elif A == "up":
        if S < N_STATES_x:    # 到達上邊界
                S_ = S
                R = -10
        elif (S - N_STATES_x) in WALLS:    # 撞到牆壁
                S_ = S
                R = -10
        elif S == GOAL:
                S_ = "terminal"
                R = 100
        elif (S - N_STATES_x) in TREASURES:
                S_ = S - N_STATES_x
                R = 10
                TREASURES.remove(S_)
                SCORE=SCORE+1
        else:
                S_ = S - N_STATES_x    # 向上移動
                R = -1
```

下:
情況 1：當下位置已經在下面邊界，S 不走動(S_ = S)，且 R =-10
情況 2：當下位置在牆的上面，S 不走動(S_ = S)，且 R =-10
情況 3：當下位置在終點的上方，S 往下走即終點，R=100
情況 4：當下位置已經在終點，最後位置就是 terminal，R=100
情況 5：往下走一步的位置是有寶藏的，S 可以往上(S_=S+N_STATES_x)，且 R=10，把得過得寶藏從寶藏陣列去除，並且得 1 分
情況 6：上述情況外，直接下走(S_= S+N_STATES_x)，R=-1

```
elif A == "down":
        if S >= (N_STATES_y - 1) * N_STATES_x:    # 到達下邊界
                S_ = S
                R = -10
        elif (S + N_STATES_x) in WALLS:    # 撞到牆壁
                S_ = S
                R = -10
        elif    (S + N_STATES_x) == GOAL:
                S_ = "terminal"
                R = 100
        elif S == GOAL:
                S_ = "terminal"
                R = 100
        elif (S + N_STATES_x) in TREASURES:
                S_ = S + N_STATES_x
                R = 10
                TREASURES.remove(S_)
                SCORE=SCORE+1
        else:
                S_ = S + N_STATES_x    # 向下移動
                R = -1

return S_, R
```

- **心得**

雖然這次的 todo 很少，但我覺得這次也不簡單，我參考很多網路上許多人設定 R 的方法，多次調整出這個最好的結果，但我要嘛是走很快到了終點，但沒有任何一個寶藏，不然就是走超多步，但有拿到寶藏。最後抉擇下，選擇了走 377 步，拿 3 個寶藏，彷彿做不到最好 QQ