1(a)

由於 output 太多，這邊只截最前面跟最後面的 iteration 當作代表，中間的部分大同小異

```
iter 0 result 100
iteration 4 move 22 goal reached
iteration 6 move 36 goal reached
iteration 9 move 40 goal reached
iteration 29 move 40 goal reached
iteration 32 move 36 goal reached
iteration 35 move 36 goal reached
iteration 38 move 24 goal reached
iteration 39 move 34 goal reached
iteration 49 move 38 goal reached
iteration 50 move 40 goal reached
iteration 51 move 28 goal reached
iteration 53 move 36 goal reached
iteration 55 move 36 goal reached
iteration 58 move 20 goal reached
iteration 61 move 46 goal reached
iteration 65 move 52 goal reached
iteration 71 move 42 goal reached
iteration 77 move 44 goal reached
iteration 92 move 34 goal reached
iteration 95 move 40 goal reached
iter 100 result 20
```

```
iteration 901 move 26 goal reached
iteration 904 move 28 goal reached
iteration 912 move 42 goal reached
iteration 917 move 28 goal reached
iteration 918 move 32 goal reached
iteration 921 move 40 goal reached
iteration 923 move 42 goal reached
iteration 927 move 50 goal reached
iteration 934 move 40 goal reached
iteration 938 move 30 goal reached
iteration 941 move 44 goal reached
iteration 942 move 24 goal reached
iteration 952 move 34 goal reached
iteration 953 move 36 goal reached
iteration 954 move 40 goal reached
iteration 956 move 22 goal reached
iteration 957 move 30 goal reached
iteration 960 move 28 goal reached
iteration 962 move 44 goal reached
iteration 963 move 24 goal reached
iteration 964 move 34 goal reached
iteration 968 move 30 goal reached
iteration 976 move 40 goal reached
iteration 977 move 34 goal reached
iteration 979 move 46 goal reached
iteration 980 move 26 goal reached
iteration 981 move 26 goal reached
iteration 983 move 26 goal reached
iteration 986 move 36 goal reached
iteration 989 move 34 goal reached
iteration 990 move 30 goal reached
iteration 991 move 36 goal reached
iteration 994 move 24 goal reached
iteration 995 move 44 goal reached
iteration 999 move 26 goal reached
```

(其中’X’代表足跡、’#’代表障礙物)

```
index 0 movement 1
index 1 movement 1
index 2 movement 2
index 3 movement 2
index 4 movement 2
index 5 movement 2
index 6 movement 2
index 7 movement 1
index 8 movement 1
index 9 movement 1
index 10 movement 2
index 11 movement 1
index 12 movement 3
index 13 movement 1
index 14 movement 2
index 15 movement 1
index 16 movement 2
index 17 movement 2
index 18 movement 2
index 19 movement 1
UP=9, RIGHT=10, LEFT=1
```

```
.........X
......XXXX
.....XX...
.....XX...
.....XX...
.....X....
#####X.###
XXXXXX....
X.........
X.........
```

1(b)

```
iter 0 result 100
iter 100 result 100
iter 200 result 100
iter 300 result 100
iter 400 result 100
iter 500 result 100
iter 600 result 100
iter 700 result 100
iter 800 result 100
iteration 842 move 31 goal reached
iteration 876 move 31 goal reached
iter 900 result 31
```

```
index 0 movement 1
index 1 movement 2
index 2 movement 3
index 3 movement 1
index 4 movement 2
index 5 movement 1
index 6 movement 2
index 7 movement 2
index 8 movement 1
index 9 movement 1
index 10 movement 2
index 11 movement 1
index 12 movement 3
index 13 movement 1
index 14 movement 3
index 15 movement 1
index 16 movement 3
index 17 movement 2
index 18 movement 2
index 19 movement 3
index 20 movement 1
index 21 movement 2
index 22 movement 3
index 23 movement 2
index 24 movement 3
index 25 movement 1
index 26 movement 3
index 27 movement 1
index 28 movement 3
index 29 movement 3
index 30 movement 3
UP=11, RIGHT=9, DOWN=11
```

```
.....#....
.....#....
.....#....
.....#....
.....X....
....XXXXX.
....X#.XXX
..XXX#...X
XXX..#...X
XX...#...X
```

1(c)

```
iter 0 result 100
iter 1000 result 100
iter 2000 result 100
iteration 2375 move 21 goal reached
iter 3000 result 21
iteration 3515 move 53 goal reached
iter 4000 result 21
iteration 4851 move 37 goal reached
iter 5000 result 21
iteration 5722 move 29 goal reached
iteration 5861 move 45 goal reached
iter 6000 result 21
iteration 6169 move 65 goal reached
iter 7000 result 21
iteration 7328 move 35 goal reached
iteration 7754 move 57 goal reached
iter 8000 result 21
iteration 8880 move 37 goal reached
iter 9000 result 21
```

2(a) s1

2(b)

```
States: 0 0 3 0 0 0 0 0 1
States: 0 0 1
States: 0 0 3 1
States: 0 0 0 0 0 0 0 0 0 0 0 0 0 1
States: 0 0 0 0 0 0 0 0 3 1
States: 0 1
States: 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 3 0 3 1
States: 0 0 3 0 0 0 0 0 0 1
States: 0 0 0 0 0 1
States: 0 0 3 0 0 0 1
```

3(a)

State	action
S0	A0
S1	A2
S2	A1

3(b)

State	action
S0	A2
S1	A2
S2	A1

3(c)

Gamma=0.8, Q-value:

```
[[ 15.90909091  12.72727273  10.18181818]
 [  0.                -inf -13.3201581 ]
 [        -inf  45.84980237         -inf]]
```

Gamma=0.95, Q-value:

```
[[21.73304188 20.63807938 16.70138772]
 [ 0.95462106        -inf  1.01361207]
 [       -inf 53.70728682        -inf]]
```

State	Action(Gamma=0.8)	Action(Gamma=0.95)
S0	A0	A0
S1	A0	A2
S2	A1	A1

S1 的 optimal policy 從 a0 變為 a2

因為 gamma 增加 => 留得住的 reward 也變多，因此 gamma=0.8 時，在 s1 原地打轉會是最佳選擇；當 gamma=0.95 時，s1 選擇 a2 走到 s2 被扣 50，s2 選擇 a1 走到 s0 得到 40 reward，s0 選擇 a0 回到原地得到 10 reward，留住的 reward 會比在 s1 原地打轉更多