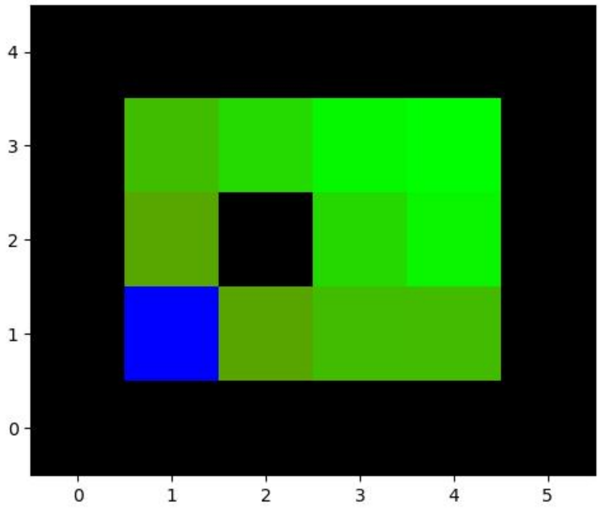
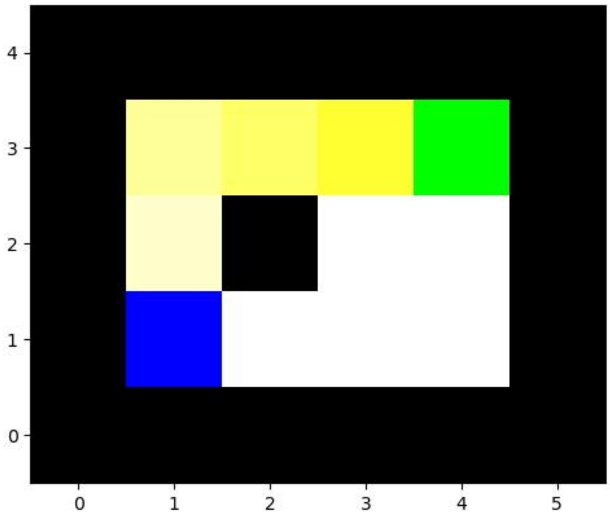
(a)





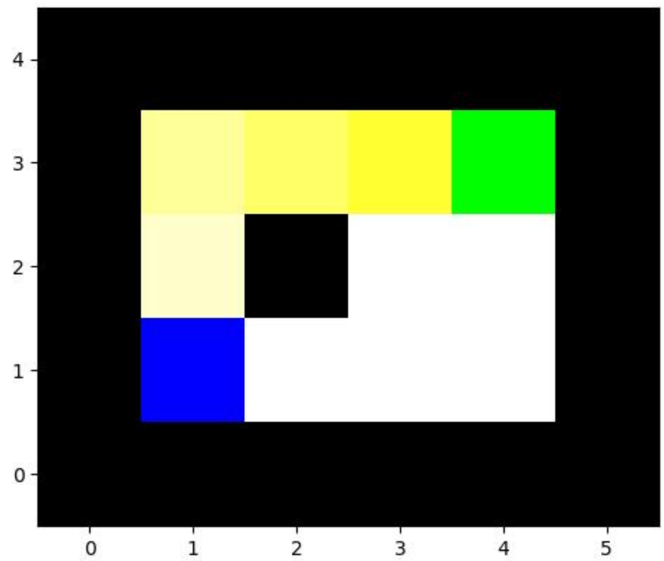
I initialize values as one. It iterates 67 times, spends 0.04 seconds.

```
The iterations: 67
Cost time is 0.048010826110839844
```

Different initialization has different iterations. If the initialization is close to the optimal value, it would have less iterations.

(b)

Same as (a)

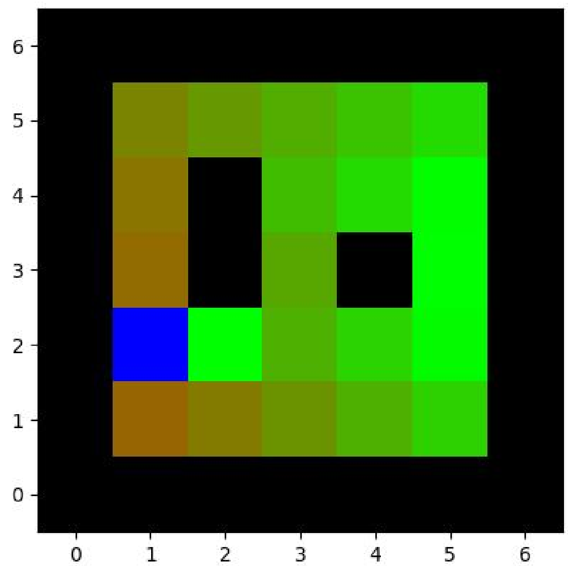


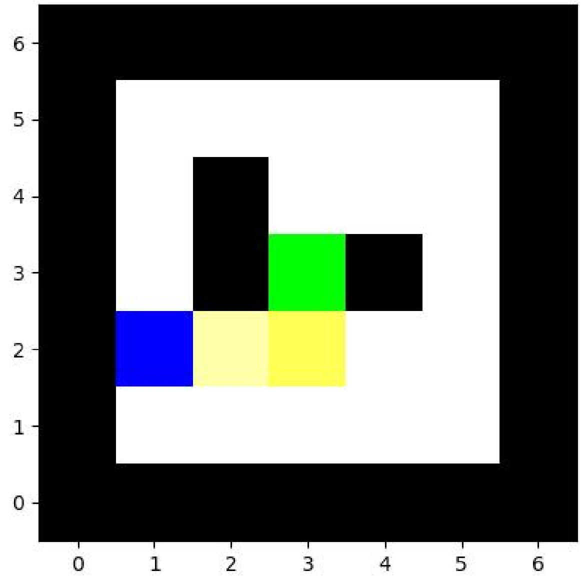
Using policy iterations, with the same initialization as part (a), it only needs 4 iterations, spends less time to finish. The policy is as same as (a)

```
The iterations: 4
Cost time is 0.024005413055419922
```

(C)

I set gamma =0.9, eta = 0.2.

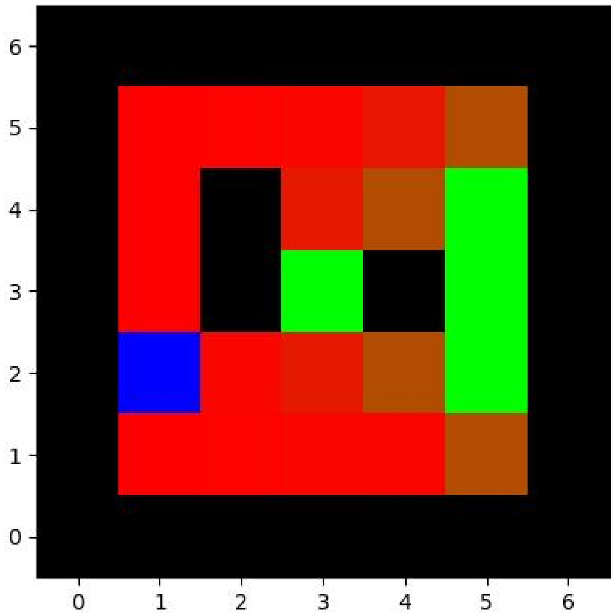


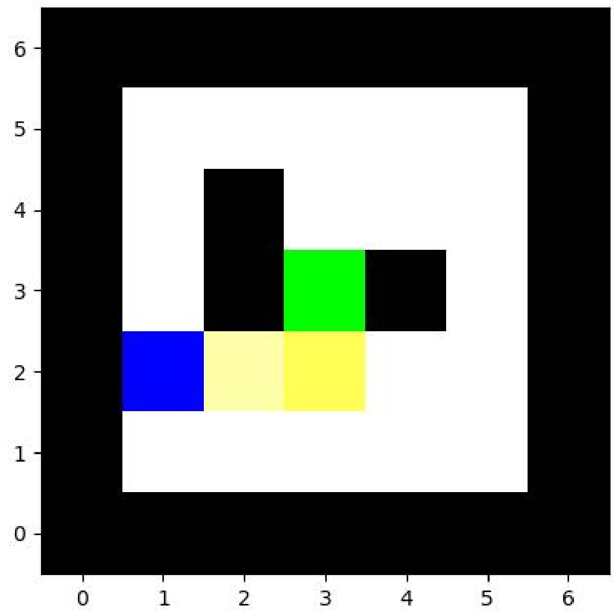


(d1)

Given that the close exit has more cost (-1) than the distant exit (-10), also prefer risking the cliff with cost +10. Thus we should decrease the importance of values V(s') in the equation, which means it is better to have a small gamma. Also we need to set eta value a bit small, to make sure it would follow the path we want.

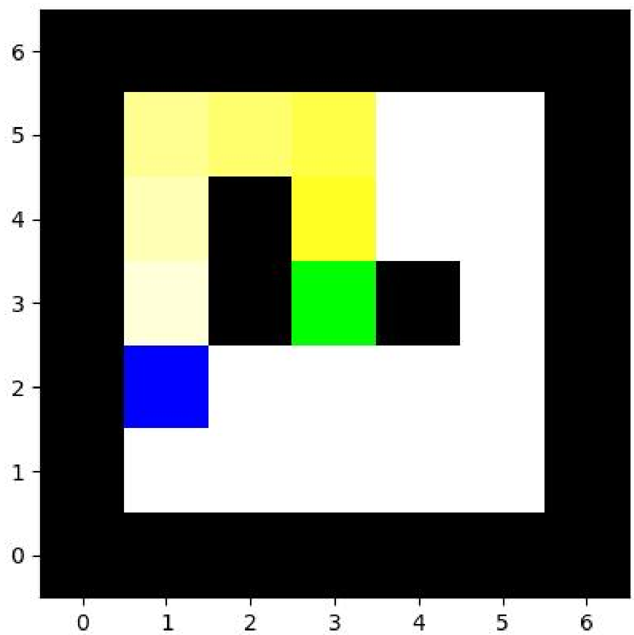
Hence I set gamma = 0.3, eta = 0.





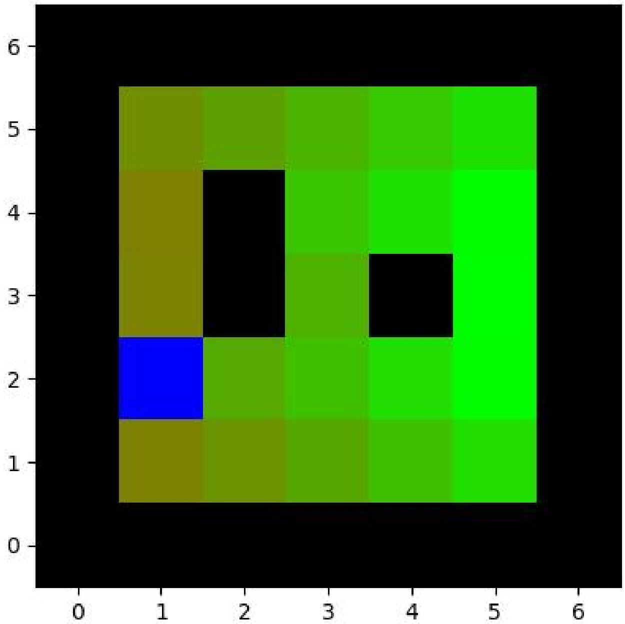
(d2)

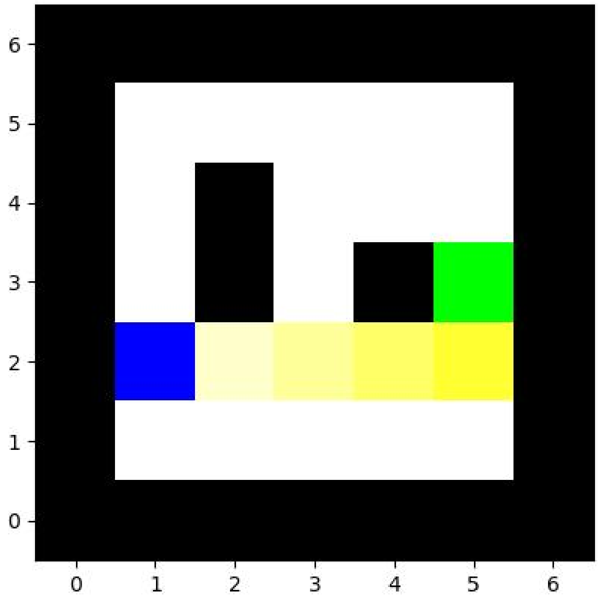
This time we need to let it avoid the cliff but exit from the same state as (d1). Thus we only need to set a noise value eta. Hence I set gamma = 0.3, eta = 0.2.



(d3)

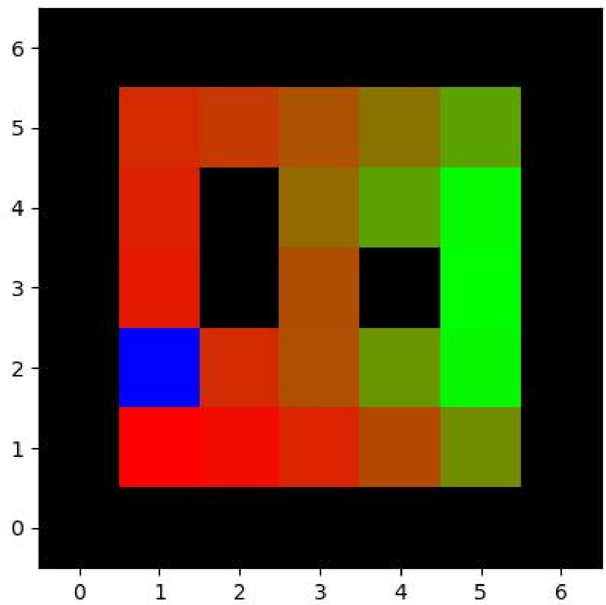
This time we need it to go from the distant exit. Thus the gamma should be a bit bigger than previous parts, make the V(s’) be important in decision. Also we need to let it risk the cliff, thus the eta should not be too small. Hence I set gamma = 0.9, eta = 0.3.



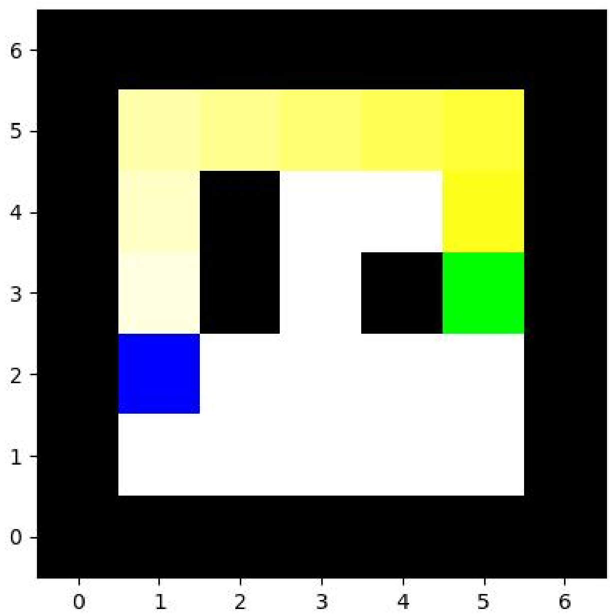


(d4)

This time we let it avoid the cliff. Thus we only need to have a small eta. I set gamma = 0.7, eta = 0.2.



0
1
2
3
4
5
6



0
1
2
3
4
5
6