Assignment 1

211220166 王诚昊

1.

动作值函数：

$$
\begin{aligned}
q_\pi(1|1) &= 1.0 \times (-1 + v_\pi(1)) \\
q_\pi(2|1) &= 0.9 \times (-1 + v_\pi(4)) + 0.1 \times (-1 + v_\pi(1)) \\
q_\pi(3|1) &= 1.0 \times (-1 + v_\pi(1)) \\
q_\pi(4|1) &= 0.9 \times (-1 + v_\pi(2)) + 0.1 \times (-1 + v_\pi(1))
\end{aligned}
$$

状态值函数：

$$
\begin{aligned}
v_\pi(1) &= \sum_{a=1}^{4} \pi(a|1) \times q(a|1) \\
&= 0.25 \times 1.0 \times (-1 + v_\pi(1)) + \\
&\quad 0.25 \times 0.9 \times (-1 + v_\pi(4)) + \\
&\quad 0.25 \times 0.1 \times (-1 + v_\pi(1)) + \\
&\quad 0.25 \times 1.0 \times (-1 + v_\pi(1)) + \\
&\quad 0.25 \times 0.9 \times (-1 + v_\pi(2)) + \\
&\quad 0.25 \times 0.1 \times (-1 + v_\pi(1))
\end{aligned}
$$

2.

1. 原代码运行结果：

```
PS D:\MyProgramming\python\UAV> python .\Assignment1.py
v(1)=75.34,v(2)=82.58,v(3)=0,v(4)=72.38,v(5)=76.13,v(6)=82.23,v(7)=69.93,v(8)=72.09,v(9)=74.89,
v(1)=97.79,v(2)=98.89,v(3)=0,v(4)=96.68,v(5)=97.77,v(6)=98.89,v(7)=95.57,v(8)=96.68,v(9)=97.78,
PS D:\MyProgramming\python\UAV>
```

2. 修改代码：

   修改键值对定义：

```python
# returns = {k: [] for k in range(1, 10)}
returns = {(x,y): [] for x in range(1, 10) for y in range(1,5)}
```

   变量是之前4倍，模拟次数也乘4很合理吧：

```python
# for episode in range(10000):
for episode in range(40000):
```

```python
# state_seq.append(s)
state_seq.append((s,a))
```

   改善输出格式：

```
#to make 4 output per line
if cnt > 3:
    print()
    cnt = 0
cnt += 1
```

运行结果：

```
PS D:\MyProgramming\python\UAV> python .\Assignment1.py
q((1, 1))=74.08,q((1, 2))=71.67,q((1, 3))=74.24,q((1, 4))=80.49,
q((2, 1))=81.31,q((2, 2))=75.63,q((2, 3))=75.03,q((2, 4))=97.21,
q((3, 1))=0,q((3, 2))=0,q((3, 3))=0,q((3, 4))=0,
q((4, 1))=73.70,q((4, 2))=69.30,q((4, 3))=71.46,q((4, 4))=74.80,
q((5, 1))=80.75,q((5, 2))=71.84,q((5, 3))=71.40,q((5, 4))=80.62,
q((6, 1))=97.25,q((6, 2))=74.48,q((6, 3))=75.85,q((6, 4))=81.42,
q((7, 1))=71.12,q((7, 2))=69.25,q((7, 3))=69.20,q((7, 4))=71.19,
q((8, 1))=74.71,q((8, 2))=71.27,q((8, 3))=69.45,q((8, 4))=73.78,
q((9, 1))=80.37,q((9, 2))=73.84,q((9, 3))=71.48,q((9, 4))=73.91,
PS D:\MyProgramming\python\UAV>
```

3. 对于确定性策略$best-policy$，蒙特卡罗方法能够有效评估状态值函数V_π，但是不能够有效评估动作值函数Q_π。因为在确定性策略下某些动作（$\frac{3}{4}$的动作）完全不会执行，因而缺少(s,a)的样本。
4. 根据第二问输出，改进后的策略如下：

```
def best_policy(s):
    p = {
        1: 4, 2: 4, 3: 4,
        4: 4, 5: 1, 6: 1,
        7: 4, 8: 1, 9: 1
    }
    return p[s]
```