Q1

`mdpVI.py` and `mdpPI.py` has the same output for all 3 rewards. Thus I only list the output once in the following results.

reward=-2

- policy

  1, 1, u 1, 2, r 1, 3, u 1, 4, u 2, 1, u 2, 3, u 2, 4, u 3, 1, r 3, 2, r 3, 3, r 3, 4, u

- value and policy graph

```
-2.91 -0.46  1.54  1.00   r r r u
-4.91       -0.62 -1.00   u   u u
-6.94 -5.25 -3.25 -2.32   u r u u
```

reward=-0.2

- policy

  1, 1, u 1, 2, r 1, 3, u 1, 4, l 2, 1, u 2, 3, u 2, 4, u 3, 1, r 3, 2, r 3, 3, r 3, 4, u

- value and policy graph

```
1.47  1.71  1.91  1.00  r r r u
1.27        1.34 -1.00  u   u u
1.03  0.84  1.04  0.84  u r u l
```

reward=-0.01

- policy

  1, 1, u 1, 2, l 1, 3, l 1, 4, l 2, 1, u 2, 3, u 2, 4, u 3, 1, r 3, 2, r 3, 3, r 3, 4, u

- value and policy graph

```
1.93  1.94  1.95  1.00 r r r u
1.92        1.55 -1.00 u   u u
1.91  1.90  1.89  1.88 u l l l
```

Q2

Note for a leaf node with equal postive and negative training samples, we can label the leaf node as either T or F. The choice will influence the LOOCV Test Error Rate for depth 2, 3 and 4. The following table is the result with choice of F when there are equal postive and negative training samples in the leaf node.

depth	LOOCV Train Error Rate	LOOCV Test Error Rate
1	0.1667	0.1667
2	0.1439	0.4167
3	0.0530	0.4167
4	0.0	0.4167

From above table, we can see that LOOCV classification error rates for both training is steadily decreasing as we increasing depth. It even decreases to 0 as we increase depth to 4.

However, the LOOCV classification error rates for test data increases as we increase depth from 1 to 2 and keep the same for depth 3 and 4.

With decreasing training error and increasing test error, this implies that overfitting is happening as we increase the depth. This is because our training data size is very small, even a depth of 2 can lead to overfitting.