lab5 Markov Decision Process

TODO 1

```
# 给定一条序列,计算从某个索引（起始状态）开始到序列最后（终止状态）得到的回报
def compute_return(start_index, chain, gamma):
    G = 0
    for i in reversed(range(start_index, len(chain))):
        # TODO ~1: 实现回报函数
        G = gamma * G + rewards[chain[i] - 1]
    return G
```

参考多项式的计算，比较简单

TODO 2

状态转移概率矩阵P表示从所有的状态s到所有的后续状态s'的转移概率

好像没什么要写的？就是改一改参数，运行一下试一试

TODO 3

```
# 对所有采样序列计算所有状态的价值
def MC(episodes, V, N, gamma):
    for episode in episodes:
        G = 0
        for i in range(len(episode) - 1, -1, -1):  # 一个序列从后往前计算
            (s, a, r, s_next) = episode[i]
            # TODO ~3: 代码填空
            G = G * gamma + r
            N[s] += 1
            V[s] += (G - V[s]) / N[s]
```

G的计算参考TODO 1，N[s]和V[s]参考slides中的公式即可