# 强化学习算法---动态规划算法分析

本实验分析” 动态规划算法.ipynb”算法。理解：

1. 怎样制作一个有模型的游戏环境？
2. 理解价值评价过程

for s in range(self.env.ncol \* self.env.nrow):  
 qsa\_list = []  
 for a in range(4):  
 qsa = 0  
 res = self.env.P[s][a][0] #从列表中提取元组即可，原来的for没有心要，因为只有一个元素  
 p, next\_state, r, done = res  
 qsa += p \* (r + self.gamma \* self.v[next\_state] \* (1-done)) # 本章节环境比较特殊，奖励和下一个状态有关，所以需要和状态转移概率相乘  
 qsa\_list.append(self.pi[s][a] \* qsa)  
 new\_v[s] = sum(qsa\_list) # 状态价值函数和动作价值函数之间的关系

不完全对应于：

1. 策略提升

for a in range(4):  
 qsa = 0  
 for res in self.env.P[s][a]:  
 p, next\_state, r, done = res  
 qsa += p \* (r + self.gamma \* self.v[next\_state] \* (1-done))  
 qsa\_list.append(qsa)  
maxq = max(qsa\_list)  
cntq = qsa\_list.count(maxq)  
self.pi[s] = [1/cntq if q == maxq else 0 for q in qsa\_list] # 让相同的动作价值均分概率

对应于：