****

**人工智能-MC and TD**



**学 院 软件学院**

**年 级 2014级**

**班 级 二 班**

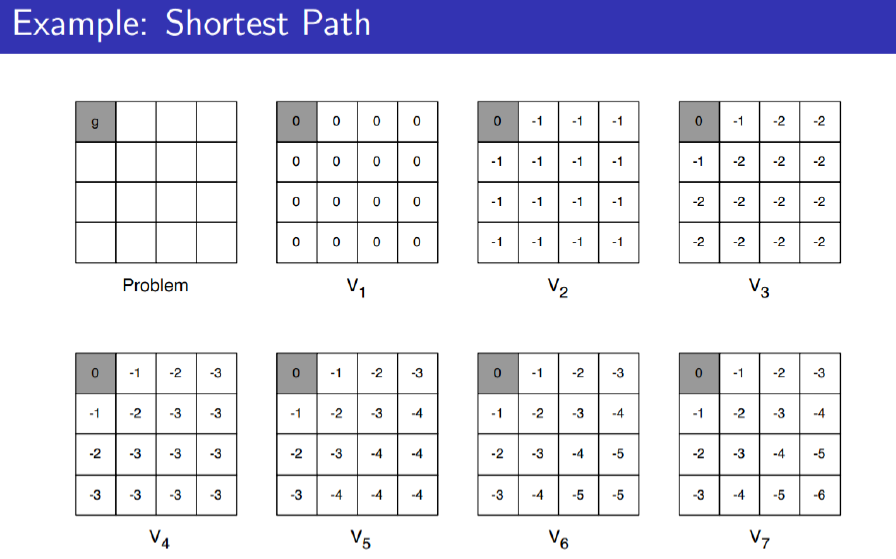
**学 号 3014218084**

**姓 名 陈恒**

**2017年 4月 16 日**

**一、实验问题**

求解某两点之间的最短路径，或是在图中加一些障碍物，计算最短路径，在开始时指定，分别使用 MC 和 TD 学习算法计算在该策略下的 v 值

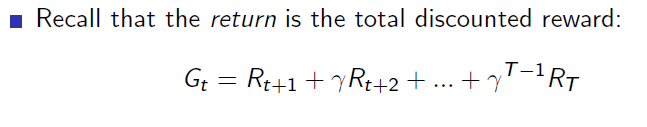


**二、实验内容**

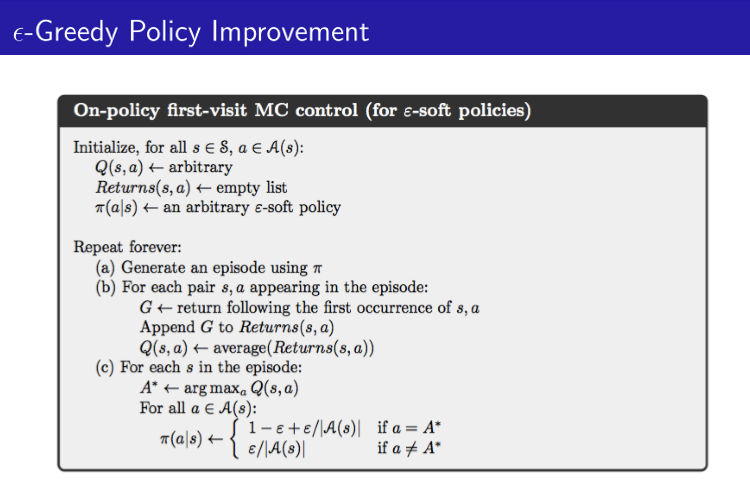
分别使用MC算法，TD算法求解在使用ϵ-Greedy更新策略下的V值，令一个状态s选择一个行动的回报r 为-1，则最终收敛的V矩阵中的V[i][j]值即可近似为该点到终点的最短距离

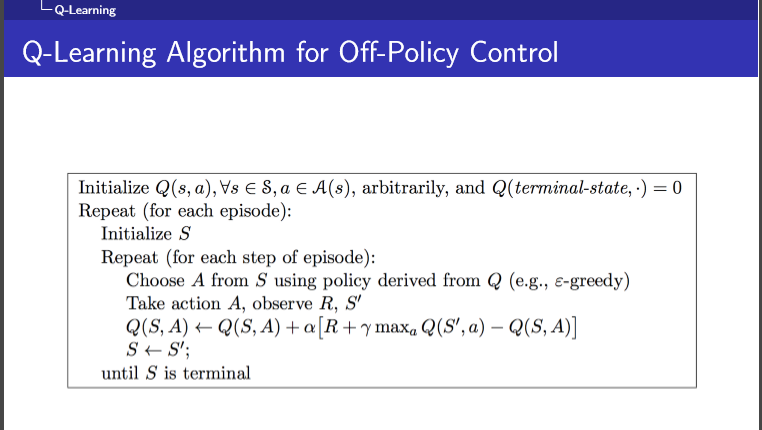
**三、实验步骤**

1.MC方法：核心思想是利用多个完整的Episodes, 对单个episode中的状态计算Gt,公式如下：



在这里使用MC相关的ϵ-Greedy Policy Improvement 算法，算法为不完全评估，即每完成一个episode对Q(s,a)的迭代，就更新一次策略的概率，并且策略的更新使用了ϵ-Greedy，目的就是能够更好的探索整个状态空间，算法伪代码如下：

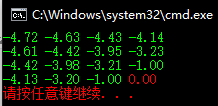
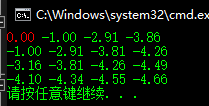


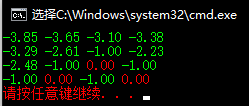
* 2.TD算法：与MC的不同之处在于，MC使用准确完整的episode来更新，而TD算法不需要完整的episode，每一步都可以更新，也就是online learning，学习快，并且适用于没有结果的场景，应用范围较广，本次实验采用与TD相关的Q-learning算法，同样使用ϵ-Greedy来对策略进行迭代，算法的伪代码如下：
* 

**四、实验结果**

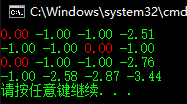
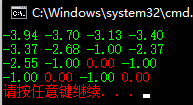
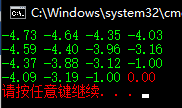
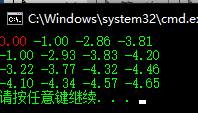
**1. MC**：

1）在不同结束点下的表现（10000次循环）：

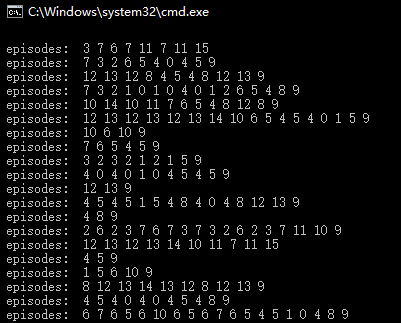




2）在不同结束点下的表现（1000次循环）：

3）实际上MC算法在实验中收敛得并不是很完美，收敛较慢，而且虽然有收敛趋势，但与真实值相差还是比较大，刚开始是百思不得其解的，后来发现有可能是随机生成序列的原因，由于C++中rand()的函数产生的是伪随机数，并且即使使用srand(time(0))设置种子值，在固定区间中产生每个数的概率并不均等，这可能是造成误差较大的原因,根据ϵ-Greedy产生的一组episodes示例：



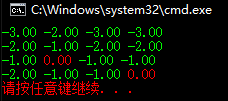
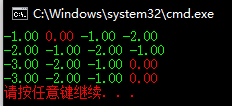
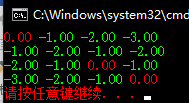
Episode结点的选择方式为： 1）随机产生一个0到1之间的double值，

2）判断产生值落在决策π的哪一个action上

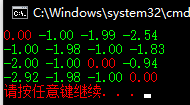
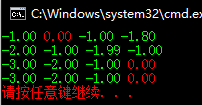
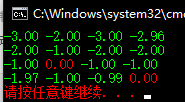
3）返回该action

**2. TD：**

1）在不同结束点下的表现（1000次循环）：

2）在不同结束点下的表现（100次循环）：



3）从上面两图中可以看出TD所用收敛次数较MC算法少，并且收敛很快，这得益于TD算法的Online特性，实时更新，快速缩小与真实值的差距。