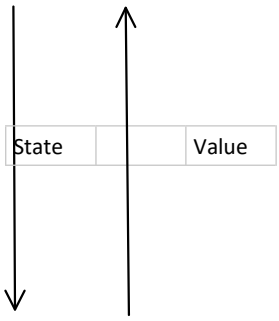
Project 2 Recitation

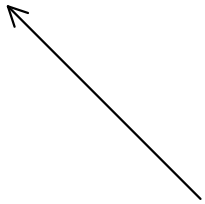
Friday, October 12, 2018 3:01 PM

.

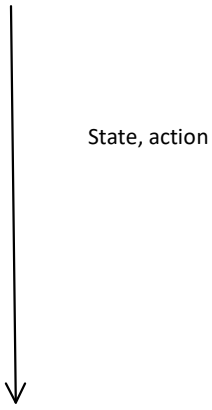
Init -> Create a dictionary (self.values)
-> Fill out the dictionary based off of previous runs
-> Loop over the number of iterations
-> Loop over all the states
-> Replace all the values in self.values with Helper[1]
-> Create a copy of self.values (.copy())



CAFV -> Return Helper()[0]



Helper -> Return a (Action, Value) (maxarg (a))
-> Take in a state, Get all actions
-> Loop over all the actions
-> If state has no actions (None, 0.0)



CQVFV -> mdp.getTransitonStatesAndProbs
-> R(s) + discount(sum(T(s,a,s')) * U(s')

Question 4

Init -> Create a dictionary (self.whatever = util.Counter)

Init -> Create a dictionary (self.whatever = util.Counter)

GetQValue -> return self.whatever[(state,action)]

CVFQV -> given a state, get the list of action
-> Iterate over all the actions (edge case of no actions)
-> Return 0.0 if no actions
-> Call GetQValue on all (state,action) pairs and take the max

CAFQV -> given a state, list of actions
-> Iterate over all the actions
-> No actions = return none
-> Call GetQValue and return the max action
-> Potential for multiple equivalent bests actions
-> Choose a random of those best actions (random.choice())

Update -> self.whatever[(state,action)] +=
-> self.alpha * (reward + (self.discount * self.getValue(nextState)) - self.whatever[(state,action)]

Question 5
GetAction -> return getPolicy(state) (CAFQV)
-> self.epsilon

-> util.coinFlip(epsilon) (same as below, eval probability)
-> random.random() < self.epsilon -> Take a random action
-> Random action is random.choice(legalActions)
-> What happens when no actions -> return None