mdp.py (original)

```
# mdp.py
# ------
# Licensing Information: Please do not distribute or publish solutions to this
# project. You are free to use and extend these projects for educational
# purposes. The Pacman AI projects were developed at UC Berkeley, primarily by
# John DeNero (denero@cs.berkeley.edu) and Dan Klein (klein@cs.berkeley.edu).
# For more info, see http://inst.eecs.berkeley.edu/~cs188/sp09/pacman.html

import random

class MarkovDecisionProcess:

  def getStates(self):
    """
    Return a list of all states in the MDP.
    Not generally possible for large MDPs.
    """
    abstract

  def getStartState(self):
    """
    Return the start state of the MDP.
    """
    abstract

  def getPossibleActions(self, state):
    """
    Return list of possible actions from 'state'.
    """
    abstract

  def getTransitionStatesAndProbs(self, state, action):
    """
    Returns list of (nextState, prob) pairs
    representing the states reachable
    from 'state' by taking 'action' along
    with their transition probabilities.

    Note that in Q-Learning and reinforcment
    learning in general, we do not know these
    probabilities nor do we directly model them.
    """
    abstract

  def getReward(self, state, action, nextState):
    """
    Get the reward for the state, action, nextState transition.

    Not available in reinforcement learning.
    """
    abstract

  def isTerminal(self, state):
    """
    Returns true if the current state is a terminal state.  By convention,
    a terminal state has zero future rewards.  Sometimes the terminal state(s)
    may have no possible actions.  It is also common to think of the terminal
    state as having a self-loop action 'pass' with zero reward; the formulations
    are equivalent.
    """
    abstract
```