CS6700: Reinforcement Learning
Programming Exercise 2
(Due on October 7th)

Herd Management

For this assignment, you are a consultant to a farmer who wants to manage his herd of cows optimally. Your task is to use dynamic programming to obtain an optimal policy for managing the farmer's herd. The overall goal is is to make as much money as possible from the herd. This is a continuing problem where the cows reproduce and the herd must be maintained over a long period of time. To simplify the problem, the only actions that the farmer can take are to sell cows of different types, thus bringing in money from the cows. Specifics of the problem are given below.

The maximum size of the herd is $H = 12$. Each cow in the herd can be one of three types: *young*, *breedable*, or *old*. During each decision epoch, each type of cow yields an expected utility, r, summarized in table 1. In addition, each cow of type i can transition to a cow of type j with probability p_{ij}. These probabilities are summarized in table 2. Offspring (*young cows*) are produced according to the probability distribution shown in table 3. Only cows of type *breedable* can produce offspring. The herd cannot grow past H cows, which means that offspring cannot be produced until the herd is reduced from H.

Note that we do not have to distinguish between individual cows and are concerned only with their types. Therefore we can represent the state of the system by a 3 dimensional vector

$$s = \langle s_{young}, s_{breedable}, s_{old} \rangle$$

where s_i is the number of cows of type i present at the beginning of each decision epoch. As an example, consider a herd with 5 cows where there are 3 young cows, 1 breedable cow, and 1 old cow. This state would be represented as:

$$s = \langle 3, 1, 1 \rangle$$

The action at each step is to decide how many cows of each type the farmer should sell. Thus an action consists of a 3 dimensional vector

$$a = \langle a_{young}, a_{breedable}, a_{old} \rangle$$

where a_i is the number of cows of type i that are sold. The constraint is that $0 \leq a_i \leq s_i$ for all types i. If the decision is made to sell a cow of type i then there is a payoff c_i, summarized in table 1. Thus if m cows of type i are sold then the payoff is $m * c_i$.

Your assignment

Use dynamic programming in the following ways to find the optimal policy for the herd management problem. Note that $R^a_{ss'}$ and $P^a_{ss'}$ need to be derived from

	Cow types		
	young	*breedable*	*old*
expected utility r	0.3	0.4	0.2
expected payoff c	2	6	4

Table 1: Expected utility and payoffs for each type of cow.

i ↓ j→	young	breedable	old
young	0.9	0.1	0
breedable	0	0.75	0.25
old	0	0.15	0.85

Table 2: Cow transition probabilities, p_{ij}

the probabilities and payoffs in the the tables. (*Hint:* You need to pay special attention to transitions that can result in more than H cows!)

1. Find the optimal value function using value iteration. Turn in a graph that shows the value function, V_k, after sweep 1, sweep 10, and the final optimal value function, $V^\star$. Since the x-axis on this graph will be the state, turn the state representation into an index into a list as follows. Treat the state as a number in a base 13 representation, with s_{young} forming the most significant digit and s_{old} forming the least significant digit. As an example, the state $\langle 3, 1, 1\rangle$ would be 521 in this representation. Use $\gamma = 0.9$ for this part.

2. Find an optimal policy using policy iteration. Again turn in a plot of the value function and how it changes over time. Use $\gamma = 0.9$ for this part.

3. With your policy iteration code, explore the effect of the discount rate, γ, on the optimal policy. Do this by solving for $\gamma = 0.9, 0.5, \text{and}, 0.3$.

   (a) Repeat the policy improvement plots for part 2 for each of the values of γ.

   (b) For each of the values of gamma, show the optimal next action when the herd is in each of the following states:

      i. 4 young cows, 7 breeding cows, 1 old cow

      ii. 1 young cow, 3 breeding cows, 6 old cows

      iii. 9 young cows, 2 breeding cows, 1 old cow

If you are concerned about what happens to the cows when they are sold, I remind you that cows have guns:
`http://www.shagrat.net/Html/cows.htm`

	Number of *young* cows produced		
cow type	0	1	2
young	1.0	0.0	0.0
breedable	0.05	0.8	0.15
old	1.0	0.0	0.0

Table 3: Offspring probabilities, q_{ij}

Hints

1. Use afterstates to make $P^a_{ss'}$ smaller. Remember that the action transitions are deterministic so you can compute $P^a_{ss'}$ by first taking s to an afterstate s_a following action a and then computing the probabilities that s_a will reach s' for all states s'. This will make P be of size $|S| \times |S|$, instead of $|S| \times |S| \times |A|$, which should be a significant savings.

   The other savings of afterstates is that the probability computations (shown below) can be shared. In particular, state $s = \langle 3, 1 \rangle$ under action $a = \langle 2, 0 \rangle$ has an afterstate of $s_a = \langle 1, 1 \rangle$. State $s = \langle 2, 2 \rangle$ under action $a = \langle 1, 1 \rangle$ has the same afterstate of $s_a = \langle 1, 1 \rangle$. Thus the probability computation does not need to be repeated. This should result in significant computational savings.

2. In order to have everyone turning in the same policies and value functions, please use the following process ordering:

   (a) You have a set of cows in some state s_t

   (b) You choose action a_t

   (c) You execute action a_t to reach an afterstate s_a

   (d) You now milk the cows (where milking corresponds to the utility)

   (e) You transition the cows from one type to another

   (f) You breed the cows and achieve a new state s_{t+1}

3. $R^a_{ss'}$ does not need to be stored as a large multi-dimensional matrix. You can write it analytically as:

$$reward = \sum_i \left[(s_i - a_i)\, r_i + (a_i c_i) \right]$$

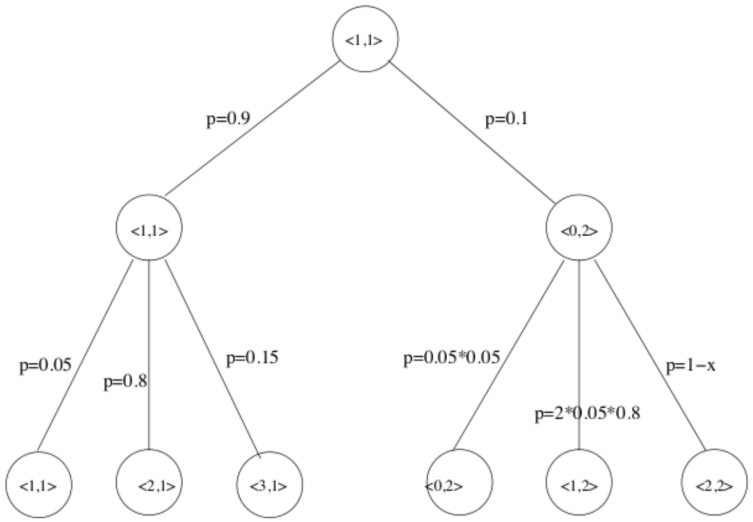
   where a_i is the ith element of the action vector (as specified above) and s_i is the number of cows that you have of type i. r_i and c_i are the utilities and payoffs specified in Table 1

4. The following is an example of how to compute $P^a_{ss'}$ for the case where you have 4 cows and two types. The type transitions for this example are:

i ↓ j→	young	breedable
young	0.9	0.1
breedable	0	1.0

The other probabilities remain as specified in the assignment.

The example state is $\langle 2, 1\rangle$. From this state, the set of valid actions is $\langle 0, 0\rangle$, $\langle 0, 1\rangle$, $\langle 1, 0\rangle$, $\langle 1, 1\rangle$, $\langle 2, 0\rangle$, and $\langle 2, 1\rangle$. For this example, let $a_t = \langle 1, 0\rangle$. This means that the afterstate, s_a, is $\langle 1, 1, \rangle$. We can compute the possible next states using the tree shown below:



In cases where you reach the maximum herd size, the final probability can be computed in two ways. You can either enumerate all the ways in which you can reach that state, or you can sum the probabilities of reaching all of the other leaves under that ancestor node. Call that sum x and the remaining probability must be $1 - x$. This is a much simpler computation and is recommended.

Using this tree, we can compute the probability of reaching a leaf state by multiplying the probabilities shown on the branches from the root to that leaf state. For example, the probability of reaching state $\langle 3, 1\rangle$ from the state $\langle 2, 1\rangle$ under action $\langle 1, 0\rangle$ is $0.9 * 0.15$. If a state appears as a leaf node more than once, add these values together.