Name: ______________________________ Andrew Id (Block Capitals): ____________________

16-731/15-780 Final, Spring 2003
*SOLUTIONS**

Thursday May 8, 2003

- You have 3 hours.
- In each question, unless we explicitly ask for an explanation, you do not need to give one.
- If you get stuck on one question, move on to others and come back to the difficult question later.
- Good luck!

Problem	Points Obtained	Points Possible
1		20
2		8
3		10
4		10
5		15
6		14
7		8
8		10
9		5
total		100

Problem 1: Short Answer Questions (20 pts)

(a) True or False: In an HMM, O_{t+1} is conditionally independent of O_{t-1} given O_t.

FALSE

(b) True or False: A key advantage of using iterative deepening search is that is uses significantly less memory than depth first search.

FALSE - They use about the same amount of memory

(c) True or False: The primary reason that matrix inversion is not (in general) used to solve neural networks is that it is too computationally expensive for large networks.

FALSE - The primary reason is that NN often use nonlinear functions that cannot be solved by matrix inversion

(d) True or False: It is not possible to use a game tree search (Min-Max) to solve a nondeterministic game.

FALSE

(e) True or False: For reinforcement learning, we need to know the transition probabilities between states before we start.

FALSE

(f) True or False: Graphplan outperforms all the other planners we discussed in class on every reasonable domain.

FALSE

(g) True or False: If, using iterated dominance, we can eliminate all strategies but one for every player, then the remaining strategies are a Nash equilibrium.

TRUE

(h) True or False: The planning algorithms we discussed in class can straightforwardly be extended to multagent domains with uncertainty.

FALSE

(i) True or False: If all the players in a game behave selfishly (e.g. play a Nash equilibrium), the outcome from a social welfare perspective is always just as good as if they had worked together.

FALSE

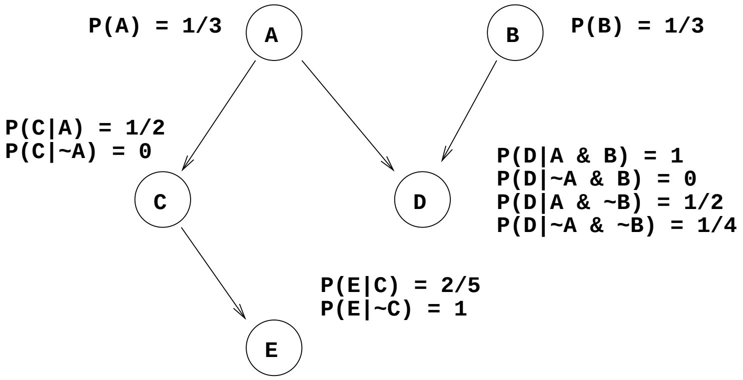
(j) True or False: Using hill-climbing search requires that you have a formula for the gradient of the function you are trying to optimize.

FALSE

(k) True or False: Consider a POMDP in which each state has only one available action. This is equivalent to a simple Markov Chain.

FALSE - There are stil hidden states, so it is an HMM

(l) Given the following Bayes Net, compute $P(B|E)$. If this takes you more than a couple of minutes you are probably not doing it the easiest available way.



B and E are D-seperated by the empty set, so P(B|E) = P(B) = 1/3

(m) The probability $P(A|B \wedge C)$ is equal to which of the following formulas? Circle the correct answer.

(i) $\frac{\sum_j P(D{=}d_j \mid A)}{P(C \mid B)P(B)}$

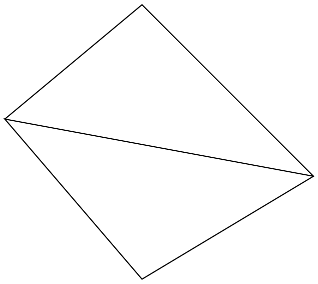
(ii) $\frac{P(C \mid B)P(B)}{P(A \wedge B \wedge C)}$

(iii) $\frac{\sum_j P(A \wedge B \wedge C \wedge D{=}d_j)}{P(C \mid B)P(B)} = \frac{P(A \wedge B \wedge C)}{P(C \wedge B)} = P(A|B \wedge C)$

(iv) $\frac{P(A \wedge B \wedge C)}{P(C \mid B)P(C)}$

(iii)

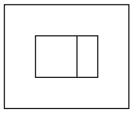
(n) Draw in the labels that the Waltz algorithm would give this figure:



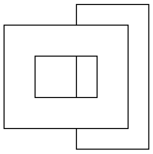
There should be arrows (pointing clockwise) on the outer edges and a + on the middel edge

(o) Among the following three drawings, circle any that the Waltz algorithm would find ambiguous.

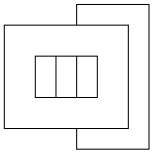
(i)



(ii)



(iii)



All three. The middle lines could be anything

(p) Learning linear regression models. Suppose you have two inputs x_1 and x_2, one output y and three weights w_0, w_1 and w_2. Your training set has 20 datapoints. You want to learn the linear model

$y = w_0 + w_1 x_1 + w_2 x + 2$

that minimizes the sum of squared residuals. You want to do this exactly with matrix algebra. Which of the following statement best describes the computational task:

(i) Solve a linear matrix equation with 2 equations in 2 unknowns

(ii) Solve a linear matrix equation with 3 equations in 3 unknowns

(iii) Solve a linear matrix equation with 20 equations in 20 unknowns

(ii)

(q) Circle all of the differences between genetic algorithms (GA) and simulated annealing (SA):

(i) GA maintains multiple candidate solutions

(ii) SA is used for minimization problems where as GA is used for maximization problems.

(iii) SA has no parameters to set whereas GA requires you to set several parameters such as the crossover rate.

(iv) GA will always converge to an optimal solution faster than SA on any given problem.

(i) only

(r) What is the entropy of these bits (examples): 0100101101100 10111

˜0.991

(s) Consider a domain where instances to be classified have $n \geq 2$ attributes. Each attribute is binary (true or false). Consider making an ”AND” decision tree that classifies instances with all the attributes true as true, and instances with at least one attribute false as false. How many leaf nodes does such a tree (minimally) have? (Give the exact number.)

n+1

(t) Now suppose we classify instances as false when they have at least TWO attributes false. Again, give an exact formula for the number of leaves (as a polynomial).

n - (n+1)/2

Problem 2: Matrix Form of Games (8 pts)

Consider the slightly modified game of rocks/paper/scissors that corrects the common misperception that paper would actually beat rock. Formally each of the two players (A and B) can select 1 of 3 actions (Rock, Paper or Scissors). Both players reveal their actions simultaneously and the winner is determined. Rock beats both scissors AND PAPER and scissors beats paper. That is if player A chooses scissors and player B choose paper, player B wins. If both players choose the same object they tie and both get a reward of 0. Otherwise the winner gets a reward of $+1$ and the loser gets a penalty of -1. The matrix form of the game is:

		Opponent		
		Rock	Paper	Scissors
You	Rock	0, 0	1, -1	1, -1
	Paper	-1, 1	0, 0	-1, 1
	Scissors	-1, 1	1, -1	0, 0

(a) Are any strategies strictly dominated? If so show the reduced matrix.

The solution is the matrix for ROCK, ROCK.

(b) Is there a pure Nash equilibrium? If so indicate it.

YES. Both players play ROCK.

Now consider the case of **s**tandard Rock/Paper/Scissors (where all the rules are identical to those above except paper beats rock). Your opponent is not a true Rock/Paper/Scissors veteran, and occasionally his hand forms rock when he means to choose something else (that is he forgets to open his hand). Specifically if he chooses paper with probability $\frac{1}{4}$ his hand forms rock instead and if he chooses scissors with probability $\frac{1}{5}$ his hand forms rock instead. You on the other hand always get the action you chose. Thus for your opponent:

P(Plays Rock | Chooses Rock) = 1
P(Plays Scissors | Chooses Rock) = 0
P(Plays Paper | Chooses Paper) = $\frac{3}{4}$
P(Plays Scissors | Chooses Scissors) = $\frac{4}{5}$
P(Plays Scissors | Chooses Paper) = 0

P(Plays Paper | Chooses Rock) = 0
P(Plays Rock | Chooses Paper) = $\frac{1}{4}$
P(Plays Rock | Chooses Scissors) = $\frac{1}{5}$
P(Plays Paper | Chooses Scissors) = 0

(c) Fill in the missing entries in the matrix form of this game. Note that your opponent's action indicates the action your opponent chooses not necessarily the one he gets.

		Opponent		
		Rock	Paper	Scissors
	Rock	0, 0	$-\frac{3}{4}, \frac{3}{4}$	$\frac{4}{5}, -\frac{4}{5}$
You	Paper	1, -1	$\frac{1}{4}, -\frac{1}{4}$	$-\frac{3}{5}, \frac{3}{5}$
	Scissors	-1, 1	$\frac{1}{2}, -\frac{1}{2}$	$-\frac{1}{5}, \frac{1}{5}$

(d) Are any strategies strictly dominated? If so show the reduced matrix.

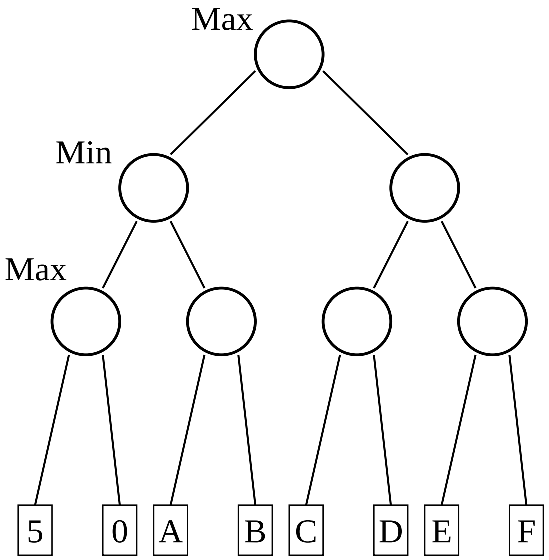
No strategies are strictly dominated.

(e) If your opponent is playing a mixed strategy of ($\frac{1}{3}$,$\frac{1}{3}$,$\frac{1}{3}$) do you want to play the same mixed strategy? If not, indicate the action you would choose the majority of the time (more than $\frac{1}{3}$rd).

No. You would want to play paper more often.

Problem 3: Game Trees (10 pts)

Consider the game tree picture below where A-F represent some real values. Assume the nodes are explored from left to right and standard alpha beta pruning is used.



(a) Give a value of A such that B is pruned.

```
Anything > 5
```

(b) Give a value of A such that B is NOT pruned.

```
Anything < 5
```

(c) **True or False:** There are SOME values of A and B such that the subtree containing C and D is pruned?

```
FALSE
```

(d) Assuming that $B = 5$ and $A = 5$, give a value of C and D such that the subtree containing E and F is pruned.

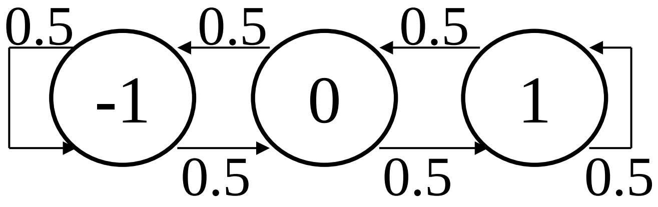
```
Anything such that max(C,B) <  5
```

(e) If you are allowed to assign A-F arbitrarily, what is the MAXIMUM number of leaves that can be pruned?

```
3 (B, E and F)
```

Problem 4: HMMs (10 pts) Consider the HMM with three states (-1,0,1) and two outputs (Y,N) given below:

$$\begin{array}{lll} pi_{-1} = 0 & pi_0 = 1 & pi_1 = 0 \\ b_{-1}(Y) = \frac{1}{4} & b_0(Y) = \frac{1}{2} & b_1(Y) = \frac{3}{4} \\ b_{-1}(N) = \frac{3}{4} & b_0(N) = \frac{1}{2} & b_1(N) = \frac{1}{4} \end{array}$$



t	$\alpha_t(-1)$	$\alpha_t(0)$	$\alpha_t(1)$
1	0	1/2	0
2	3/16	0	1/16

Note: For some questions it may be easier to express all numbers as: $\frac{X}{2^Y}$ for some X and Y.

(a) Fill in the α table (for steps 1 and 2) above if the output (Y,N) was seen.

(b) What is the probability $q_2 = 1$ (the state on timestep 2 is 1) given this output?

1/4

(c) True or False: There is NO sequence of n output symbols ($n > 1$) that would allow you to perfectly determine your state (i.e. the probability of being in some state on time step n is 1) on the above HMM.

TRUE

(d) Create a new 2 state HMM (transition probabilities, start probabilities and output probabilities = 8 numbers total) such that the probability of the observations (Y,N,Y) is 1.

$$\begin{array}{cc}
\pi_0 = 1 & \pi_1 = 0 \\
b_0(Y) = 1 & b_0(N) = 0 \\
b_1(Y) = 0 & b_1(N) = 1 \\
\delta(0,1) = 1 & \delta(0,0) = 0 \\
\delta(1,0) = 1 & \delta(1,1) = 0
\end{array}$$

(e) Create a new HMM from the one in part (d) by *only changing TWO numbers* such that the probability of the observations (Y,N,Y) is $\frac{1}{4}$. Indicate the numbers changed and the new values.

There a several solutions such as:

$\pi_0 = 1/4$ and $\pi_1 = 3/4$ OR

$b_0(Y) = 1/2$ and $b_0(N) = 1/2$

(f) **True or False:** Given a sequence of observations (O_1, ..., O_N) and any real number $R \in [0, 1]$ it is *ALWAYS* possible to create an HMM such that $P(O_1, ..., O_N) = R$.

TRUE - You can always create a chain of N states such that state i goes to i+1 with probability 1 and state i see Oi with probability 1.

Problem 5: Markov Decision Processes (15 pts)

(a) Suppose we have a system very similar to a Markov Decision Process, except that instead of trying to maximize our expected discounted delayed rewards, we wish to minimize the expected time to reach a specific state called the goal state. Each transition takes exactly one time step. Write

$$p_{ij}^a = Prob(next = j | this = i \wedge action = a)$$

and let the goal state be i_{goal}. Define

$J^*(i)$ = Expected time to goal state starting from i if we follow the optimal policy
$\pi^*(i)$ = the optimal policy (i.e. the optimal action to take at state i)

We now write down the update equations for a value iteration solution to this problem. $J^k(i)$ denotes the value for state i on the k'th iteration of value iteration. There are two bugs in the equations.

$$J^0(i) = 0 \quad \forall i$$

$$J^{k+1}(i) = \begin{cases} 0 & i = i_{\text{goal}} \\ min_a \sum_j p_{ij}^a J^k(j) & i \neq i_{\text{goal}} \end{cases}$$

$$\pi^*(i) = max_a J^*(i)$$

(where $J^*(i)$ are the values once value iteration has converged).

Your job: explain the two bugs in the boxes provided below, and then in the remaining boxes rewrite the update equations, altering them where necessary to be correct.

Bug number 1:	$min_a \sum_j p_{ij}^a J^k(j)$ will always just be 0.
Bug number 2:	$\pi^*(i) = max_a J^*(i)$ should be a min_a
	$J^0(i) = 0 \forall i$
	$J^{k+1}(i) = 1 + min_a \sum_j p_{ij}^a J^k(j)$ if $i \neq i_{\text{goal}}$ (and 0 for $i = i_{goal}$)
	$\pi^*(i) = min_a J^*(i)$

(b) In a different formulation, we want to hit the goal on an even-numbered timestep. There is a single distinguished state called i_{goal}. It is a terminal state. We get a reward of 100 dollars if we first arrive at i_{goal} on an even numbered timestep. We get a reward of -100 dollars if we first arrive at igoal on an odd-numbered timestep. We want to compute a policy that maximizes our expected reward.

Describe how you would solve this. It is not essential that you give explicit equations. Your explanation could involve one or more of:

- Write out the equations of a new value function (or set of value functions)
- You may explain how to alter the state space definition of the problem

`ANSWER:` If the original MDP had n states called 1, 2, ..., n create a new MDP with 2n states called 1e, 2e, ..., ne, 1o, 2o, ..., no.

In the new MDP

$P(next\ =\ xe\,|this\ =\ ye, a) = 0\ \forall x, y$

$P(next\ =\ xo\,|this\ =\ yo, a) = 0\ \forall x, y$

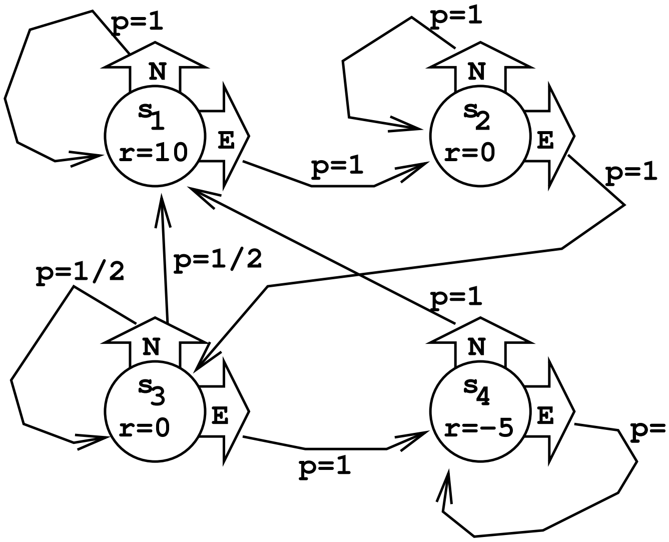
$P(next\ =\ xe\,|this\ =\ yo, a) = P_{old}(next\ =\ x|this\ =\ y, a)$

$P(next\ =\ xo\,|this\ =\ ye, a) = P_{old}(next\ =\ x|this\ =\ y, a)$

Put in an immediate reward of 0 for all states except $i_{goal}e$ (reward of 100) and $i_{goal}o$ (reward of -100). Make $i_{goal}e$ and $i_{goal}o$ terminal states.

Use a discount factor $\gamma = 1$.

(c) Consider this MDP with delayed rewards and a discount factor $\gamma = \frac{1}{2}$



Assume we have a policy π_0 which is to always go north. Define $J^{\pi_0}(i)$ = expected sum of discounted rewards if we start at state i and follow policy π_0. Write down the numerical values of:

$J^{\pi_0}(s_1) = 20$ $J^{\pi_0}(s_2) = 0$

$J^{\pi_0}(s_3) = 20/3$ $J^{\pi_0}(s_4) = 5$

(d) Continuing from part (d), suppose we run policy iteration with π_0 as the initial policy. Define π_1 = Updated policy after one iteration of policy iteration (i.e. after one policy-improvement step). Write down the values of:

$\pi_1(s_1) = \text{N}$ $\pi_1(s_2) = \text{E}$

$\pi_1(s_3) = \text{N}$ $\pi_1(s_4) = \text{N}$

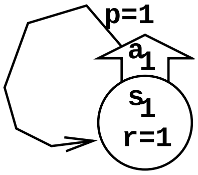
(e) Continuing on from part (e), Define π^* = Final policy after Policy iteration converges. Write down the values of:

$\pi^*(s_1) = \text{N}$ $\pi^*(s_2) = \text{E}$

$\pi^*(s_3) = \text{N}$ $\pi^*(s_4) = \text{N}$

Problem 6: Reinforcement Learning (14 pts)

(a) Consider this (rather trivial) MDP.



Suppose we decide to run Q-learning. The Q-table consists of just one entry

Q(s,a)	a = a1
s = s1	

Suppose we initialize the Q-table to zero, and then run Q-learning. Assume a discount factor γ and a learning rate α.

Let q_t = Value in the q-table after observing and processing t transitions. Note that $q_0 = 0$. Eventually q_t will converge to the true $Q^*(s1, a1)$ value.

It turns out that

$$q_{t+1} = Xq_t + Y$$

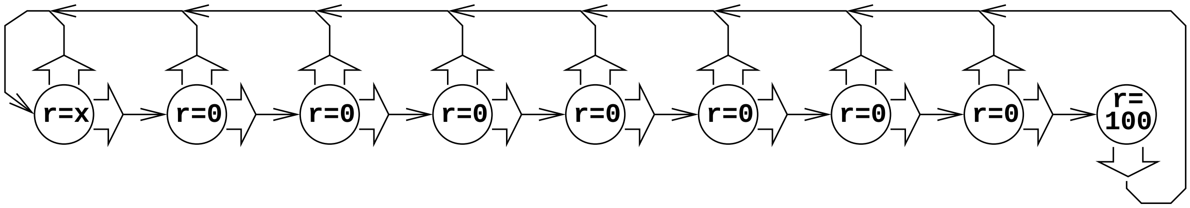
for certain values of X and Y that depend on α and γ.

Your job: Determine X and Y (each will be an expression that may involve one or both of γ and α).

`ANSWER:` We have $q_{t+1} = (1 - \alpha)q_t + \alpha(1 + \gamma q_t) = (1 - \alpha + \alpha\gamma)q_t + \alpha$.

So $X = 1 - \alpha + \alpha\gamma$ and $Y = \alpha$.

(b) Consider Q-learning with the following MDP.



Note that all actions are deterministic. Notice that if you choose the “North Action” you transition to the leftmost state. The leftmost state has immediate reward x, the rightmost has immediate reward 100 and all others have zero immediate reward. Assume a discount factor of γ = 0.99.

Assume the Q-table is initialized with all zeroes. Since the MDP is deterministic, we choose a learning rate $\alpha = 1$. Consider three exploration strategies:

ES1 = Always randomly choose an available action with 50-50 probability
ES2 = Always choose action = argmax over a of $Q(s, a)$ (with ties broken randomly)
ES3 = 99% of transitions do ES2 and 1% of the transitions do ES1

In each of the boxes below, circle the most appropriate statement:

- **FIND-OPTIMAL-QUICK:** The policy implied by the Q-table values will probably become the optimal policy before 200 state transitions have happened
- **FIND-OPTIMAL-SLOW:** The policy implied by the Q-table values will eventually become the optimal policy but it will probably take at least 200 state transitions
- **NEVER-FIND-OPTIMAL:** The policy implied by the Q-table values will probably never become the optimal policy

Important facts:

γ = 0.99
α (learning rate) = 1
Q-values initialized to zero.

“The policy implied by the Q-table” is the policy you’d follow if you always chose $argmax_a Q(s, a)$. Note that if the Q-values were correct, the policy implied by the q-table would be optimal, no matter what the exploration strategy.

x	Using ES1	Using ES2	Using ES3
0	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL
-1	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL
+1	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL	FIND-OPTIMAL-QUICK FIND-OPTIMAL-SLOW NEVER-FIND-OPTIMAL

Answer: When exploring randomly, starting out from the leftmost node, the probability of hitting the rightmost node before ending up at the leftmost node again is only 2^{-8}, so it will (in expectation) take a long time (more than 200 transitions) to even find this payoff.

So the solution is:

```
              ES1        ES2        ES3
x=0           slow       slow       slow
(all strategies are exploring randomly until
the rightmost node is found)

x=-1          slow       quick      quick
(the greedy strategies will try to stay away
from the leftmost node and
find the rightmost node)

x=1           slow       never      (really) slow
(ES2 will keep going back to the leftmost node
and never discover the reward at the rightmost node.
ES3 will discover the rightmost node at some point,
but for this an event with probability 1/200 has to
take place 8 times in a row.)
```

Problem 7: Continuous Game Theory (8 pts)

Consider a game where you and your neighbor are deciding what car to buy. Cars come in different sizes: you need to decide on the size of your car $x \in [0.5, 2]$. A car of size x will cost you exactly x. In this game, the only reason to buy a bigger car is that if you get into a crash with your neighbor, you will be better off in a bigger car (and your neighbor will be worse off). Interestingly, the expected cost of being in a crash is exactly the ratio of her car's size to your car's size. Thus we have the following utility functions:

$u_1(x_1, x_2) = -x_1 - \frac{x_2}{x_1}$

and symmetrically

$u_2(x_1, x_2) = -x_2 - \frac{x_1}{x_2}$.

(a) Write down $\frac{\partial}{\partial x_1} u_1(x_1, x_2)$.

$-1 + \frac{x_2}{x_1^2}$.

(b) Find the (symmetric) Nash equilibrium of the car buying game.

ANSWER: The best response to x_2 is given by $-1 + \frac{x_2}{x_1^2} = 0 \Rightarrow x_1 = \sqrt{(x_2)}$. (Note that the derivative is positive to the left of this point and negative to the right, so this is in fact a best response.) Similarly we must have $x_2 = \sqrt{(x_1)}$ and so, $x_1 = x_2 = 1$.

(c) If the neighbors collaborated to maximize social welfare for both of them, what would be the optimal car size? (Assume they get the same size car.)

ANSWER: $x_1 = x_2 = 0.5$. (If you both get the same car size, the ratio is always the same, so you just want to minimize the direct cost.)

Problem 8: Game Theory: Matching Pennies with a Double-heads Fetish (10 pts)

Recall the game of matching pennies, where player 1 seeks to match the pennies and player 2 seeks to have them different. However, now player 1 derives a strange additional pleasure from seeing both heads at the same time. The game thus becomes (player 1 is the row player, player 2 is the column player)

	H	T
H	2,-1	-1,1
T	-1,1	1,-1

This game has a unique (mixed strategy) Nash equilibrium. Find player 2's mixed strategy in this equilibrium.

ANSWER: For player 1 to be indifferent between playing heads and playing tails, we must have that the following are the same (where p_{2H} is the probability of player 2 playing heads)

$2p_{2H} - 1(1 - p_{2H})$ (player 1's utility of playing heads) and $-1p_{2H} + 1(1 - p_{2H})$ (player 2's utility of playing tails)

Solving this gives $p_{2H} = .4$.

For player 2 to be indifferent, player 1's strategy must be the same as in the equilibrium in the original matching pennies game (because player 2's payoffs did not change at all). Thus $p_{1H} = .5$.

Problem 9: Technologies and Applications (5 pts)

For each of the below applications/scenarios described below, indicate which technology (of: Markov Chains (MC), Markov Decision Problems (MDPs), Partially Observable Markov Decision Problems (POMDPs), Reinforcement Learning (RL), or Hidden Markov Models (HMMs)) is *best* suited.

(a) You are a spectator at an NBA playoffs game. You have good seats and can see absolutely everything that is going on in the game, but your seats are *not* close enough to the court that you can influence the game by shouting at the players. All you can do is sit and watch how the game develops.

`Markov Chain`

(b) You are chief of police in the old mafia-dominated Chicago, and you are trying to bring down gang-related crime. You have many choices in how you try to do this, e.g. how many police you station everywhere, trying to pressure witnesses to give you vital information, etc. Of course you are not perfectly aware of all the crime that is going on: all you have is indicators (how many murders, bombings, etc.).

`POMDP`

(c) You are trying to model the movements of the entire stock market. All you have is observations of the price of a single stock.

`HMM`

(d) You are playing a game of Tic Tac Toe against a random opponent. You can see the board and choose actions, but your opponent choose random actions.

`MDP`

(e) Playing the Tower of Hanoi game, except that each time you ask to make a move there's a disk-dependent probability that the disk will jump to a random position on a random stick. You don't know these probabilities in advance but you are promised that while you're practicing, the probabilities will remain fixed.

`Reinforcement learning`

(f) You're a bank trying to maximize profit from a customer. You already know an accurate statistical model of how customers change over time, and as a result of the banks actions, but you only get noisy indications of what the customer's current status is. As a bank you can offer various promotions and rewards to your customer at various times.

`POMDP`