CS 188 Introduction to Summer 2014 Artificial Intelligence

Midterm

- You have approximately 2 hours and 50 minutes.
- The exam is closed book, closed notes except your one-page crib sheet.
- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences AT MOST.

First name	
Last name	
SID	
edX username	

First and last name of student to your left	
First and last name of student to your right	

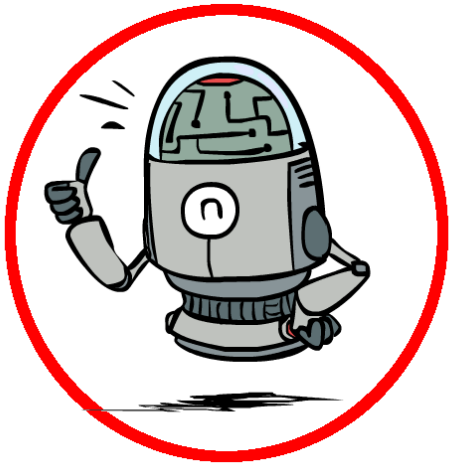
For staff use only:

Q1.	Warm-Up	/1
Q2.	Search: Algorithms	/12
Q3.	Search: Formulation	/8
Q4.	CSPs: Potluck Pandemonium	/13
Q5.	Games	/15
Q6.	MDPs: Water Slide	/7
Q7.	RL: Dangerous Water Slide	/7
Q8.	RL: Amusement Park	/12
	Total	/75

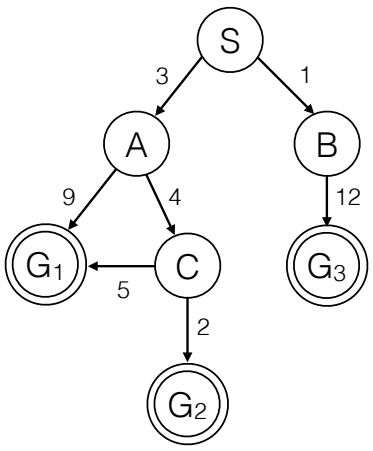
THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [1 pt] Warm-Up

Circle the CS188 mascot



Q2. [12 pts] Search: Algorithms



	A	B	C	S
H-1	0	0	0	0
H-2	6	7	1	7
H-3	7	7	1	7
H-4	4	7	1	7

(a) Consider the search graph and heuristics shown above. Select **all** of the goals that **could** be returned by each of the search algorithms below. For this question, if there is a tie on the fringe, assume the tie is broken **randomly**.

(i) [1 pt] DFS — G_1 ● G_2 ● G_3 ●

(ii) [1 pt] BFS — G_1 ● G_2 ○ G_3 ●

(iii) [1 pt] UCS — G_1 ○ G_2 ● G_3 ○

(iv) [1 pt] Greedy with H-1 — G_1 ● G_2 ● G_3 ●

(v) [1 pt] Greedy with H-2 — G_1 ● G_2 ○ G_3 ○

(vi) [1 pt] Greedy with H-3 — G_1 ● G_2 ○ G_3 ●

(vii) [1 pt] A* with H-2 — G_1 ○ G_2 ● G_3 ○

(viii) [1 pt] A* with H-3 — G_1 ○ G_2 ● G_3 ○

(b) For each heuristic, indicate whether it is consistent, admissible, or neither (select more than one option if appropriate):

(i) [1 pt] H-1 — Consistent ● Admissible ● Neither ○

(ii) [1 pt] H-2 — Consistent ○ Admissible ● Neither ○

(iii) [1 pt] H-3 — Consistent ○ Admissible ○ Neither ●

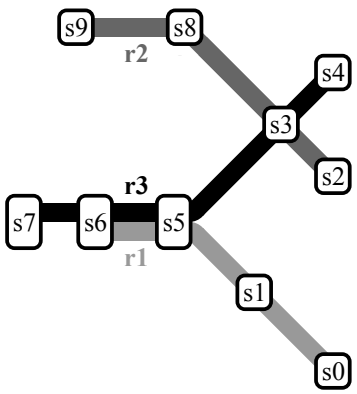
(iv) [1 pt] H-4 — Consistent ● Admissible ● Neither ○

Q3. [8 pts] Search: Formulation

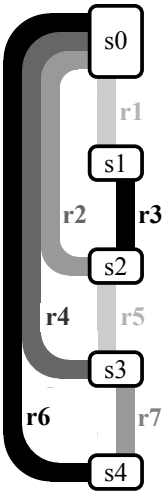
You are building an app that tells users how to travel between two locations in a subway system as cheaply as possible. You have a map that shows all of the train routes. Two example maps are shown to the right.

There are R train routes and S stations. It costs \$1 to board a train or to transfer trains. You can transfer between trains if they visit the same station (e.g. in the sparse map you can transfer from route 2 to route 3 at station 3).

To be clear, the cost does not vary for different train routes, it does not depend on where you get on or off each train, or how long you spend waiting at stations, it only depends on the number of trains you catch.



Sparse Map
9 Stations
3 Routes



Dense Map
5 Stations
7 Routes

(a) Formulate this problem as a search problem. Choose your definition of states so that the state space is small. Assume R **is larger than** S (a dense map).

(i) [1 pt] States: Each state is represented by a station, the station you are currently at.

(ii) [1 pt] Successor function: Given the current station, it returns stations that can be reached by travelling on a train route from that station. The actions have a cost of \$1.

(iii) [1 pt] Start state: The starting station.

(iv) [1 pt] Goal test: Is the current station the goal station?

(b) Now assume R **is smaller than** S (a sparse map). Formulate this problem as a search problem again, choosing a definition of states so that the state space is small.

(i) [1 pt] States: Each state is represented by a train route, the train route you are currently on.

(ii) [1 pt] Successor function: Given the current train route, it returns all train routes that share a station with that train route. The actions have a cost of \$1.

(iii) [1 pt] Start state: A special train route that visits only the start station.

(iv) [1 pt] Goal test: Does the current train route visit the goal station?

Q4. [13 pts] CSPs: Potluck Pandemonium

The potluck is coming up and the staff haven't figured out what to bring yet! They've pooled their resources and determined that they can bring some subset of the following items.

1. Pho
2. Apricots
3. Frozen Yogurt
4. Fried Rice
5. Apple Pie
6. Animal Crackers

There are five people on the course staff: Taylor, Jonathan, Faraz, Brian, and Alvin. Each of them will only bring one item to the potluck.

i. If (F)araz brings the same item as someone else, it cannot be (B)rian.

ii. (A)lvin has pho-phobia so he won't bring Pho, but he'll be okay if someone else brings it.

iii. (B)rian is no longer allowed near a stove, so he can only bring items 2, 3, or 6.

iv. (F)araz literally can't even; he won't bring items 2, 4, or 6.

v. (J)onathan was busy, so he didn't see the last third of the list. Therefore, he will only bring item 1, 2, 3, or 4.

vi. (T)aylor will only bring an item that is before an item that (J)onathan brings.

vii. (T)aylor is allergic to animal crackers, so he won't bring item 6. (If someone else brings it, he'll just stay away from that table.)

viii. (F)araz and (J)onathan will only bring items that have the same first letter (e.g. Frozen Yogurt and Fried Rice).

ix. (B)rian will only bring an item that is after an item that (A)lvin brings on the list.

x. (J)onathan and (T)aylor want to be unique; they won't bring the same item as anyone else.

This page is repeated as the second-to-last page of this midterm for you to rip out and use for reference as you work through the problem.

(a) [1 pt] Which of the listed constraints are unary constraints?

i ○ ii ● iii ● iv ● v ●

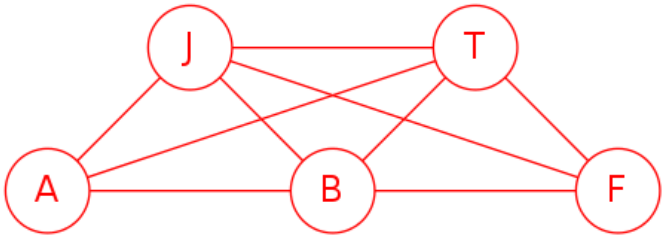
vi ○ vii ● viii ○ ix ○ x ○

(b) [2 pts] Rewrite implicit constraint viii. as an explicit constraint.

(F, J) ∈ { (3, 4), (4, 3),
(2, 5), (5, 2),
(2, 6), (6, 2),
(5, 6), (6, 5),
(1, 1), (2, 2), (3, 3),
(4, 4), (5, 5), (6, 6)
}

(c) [1 pt] How many edges are there in the constraint graph for this CSP?

There are 9 edges in this constraint graph.



(d) [1 pt] The table below shows the variable domains after all unary constraints have been enforced.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

Following the Minimum Remaining Values heuristic, which variable should we assign first? Break all ties alphabetically.

A ○ B ● F ○ J ○ T ○

(e) To decouple this from the previous question, assume that we choose to assign (F)araz first. In this question, we will choose which value to assign to using the Least Constraining Value method.

To determine the number of remaining values, enforce arc consistency to prune the domains. Then, count the total number of possible assignments (**not** the total number of remaining values). It may help you to enforce arc consistency twice, once before assigning values to (F)araz, and then again after assigning a value.

The domains after enforcing unary constraints are reproduced in each subquestion. The grids are provided as scratch space and **will not** be graded. Only numbers written in the blanks will be graded. The second grid is provided as a back-up in case you mess up on the first one. More grids are also provided on the second-to-last page of the exam.

(i) [2 pts] Assigning F = 1 results in 0 possible assignments.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

Assigning F = 1 leaves no possible values in J's domain (due to constraint viii).

(ii) [2 pts] Assigning F = 3 results in 5 possible assignments.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

Assigning F = 3 leaves J's domain as $\{4\}$. Enforcing arc consistency gives A = $\{2, 3, 5\}$, B = $\{6\}$, and T = $\{1, 2\}$. Therefore, the 5 possible assignments are (A, B, F, J, T) = (2, 6, 3, 4, 1), (3, 6, 3, 4, 1), (5, 6, 3, 4, 1), (3, 6, 3, 4, 2), (5, 6, 3, 4, 2).

(iii) [2 pts] Assigning F = 5 results in 3 possible assignments.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

Assigning F = 5 leaves J's domain as $\{2\}$. Enforcing arc consistency gives A = $\{3, 4, 5\}$, B = $\{6\}$, and T = $\{1\}$. Therefore, the 3 possible assignments are (A, B, F, J, T) = (3, 6, 5, 2, 1), (4, 6, 5, 2, 1), (5, 6, 5, 2, 1).

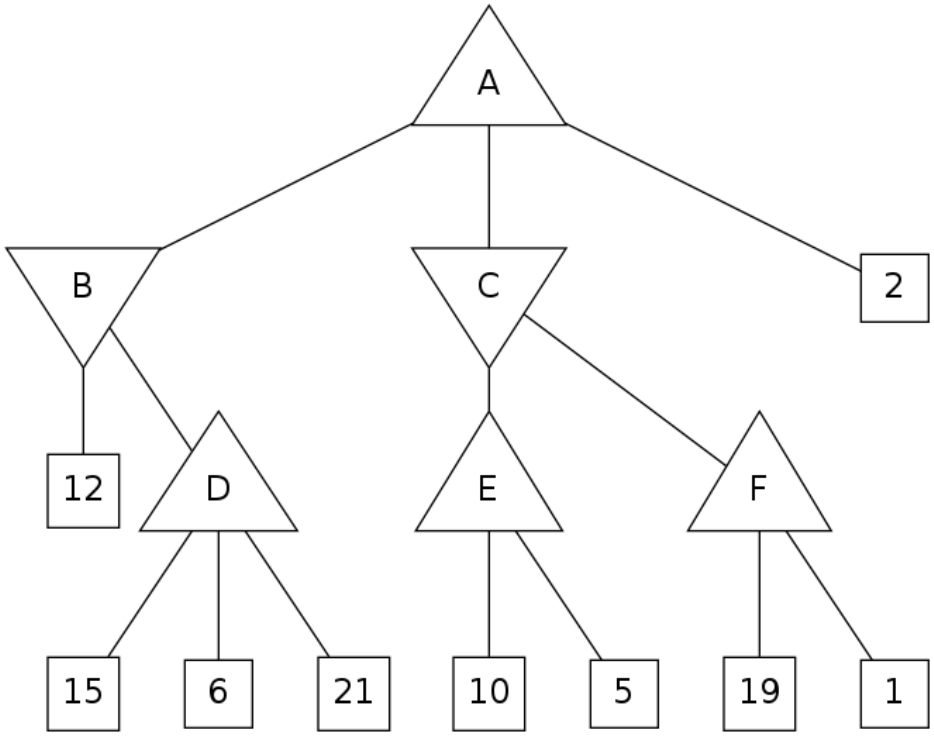
(iv) [1 pt] Using the LCV method, which value should we assign to F? If there is a tie, choose the lower number. (e.g. If both 1 and 2 have the same value, then fill 1.)

1 ○ 2 ○ 3 ● 4 ○ 5 ○ 6 ○

(f) [1 pt] We can also take advantage of the structure of this CSP to solve it more efficiently. How many variables are in the smallest cutset that will lead to a reduced CSP that is a tree? (There may be multiple cutsets of this size.)

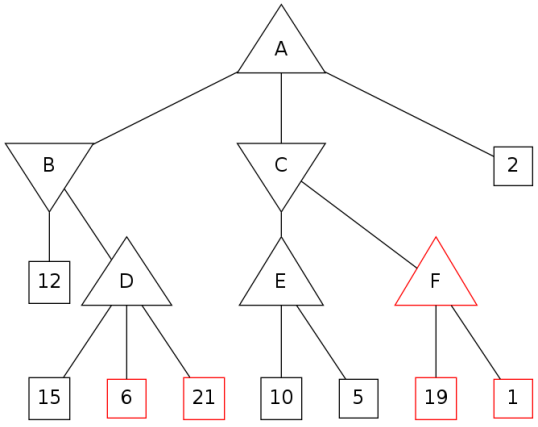
0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5 ○

Q5. [15 pts] Games



(a) [1 pt] What is the minimax value of node A in the tree above?
12

(b) [2 pts] Cross off the nodes that are pruned by alpha-beta pruning. Assume the standard left-to-right traversal of the tree. If a non-terminal state (A, B, C, D, E, or F) is pruned, cross off the entire subtree.



(c) [5 pts] If a function F is strictly increasing, then $F(a) < F(b)$ for all $a < b$ for $a, b \in \mathbb{R}$. Consider applying a strictly increasing function F to the leaves of a game tree and comparing the old tree and the new tree.

Are the claims below true or false? For true cases, justify your reasoning in a single sentence. For false cases, provide a counterexample (specifically, a game tree, including terminal values).

In a *Minimax* two player zero-sum game, applying F will not change the optimal *action*.

True, $\min_i(x_i) = \min_i(F(x_i))$ and $\max_i(x_i) = \max_i(F(x_i))$ because strictly increasing transformation doesn't change ordering.

In a *Minimax* two player zero-sum game, applying F will not affect which nodes are pruned by alpha-beta pruning.

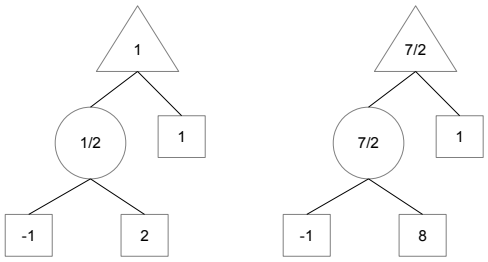
True, the alpha-beta implementation takes the same steps since the ordering on values remain the same. In other words, no inequality changes after the transformation.

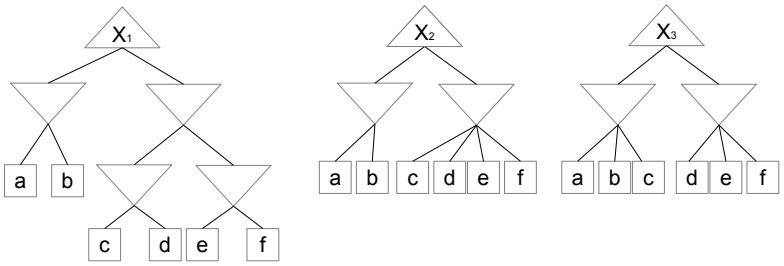
In a *Minimax* two player non-zero-sum game (where the utilities of players do not necessarily add up to zero), applying F will not change the optimal *action*.

True, again the ordering doesn't change for each player so they take the same actions. That is, if u_i was maximal for a given max node, $F(u_i)$ remains maximal after the transformation.

In an *Expectimax* two player zero-sum game, applying F will not change the optimal *action*.

False, let $F(x) = x^3$ and the chance node having equal probability.



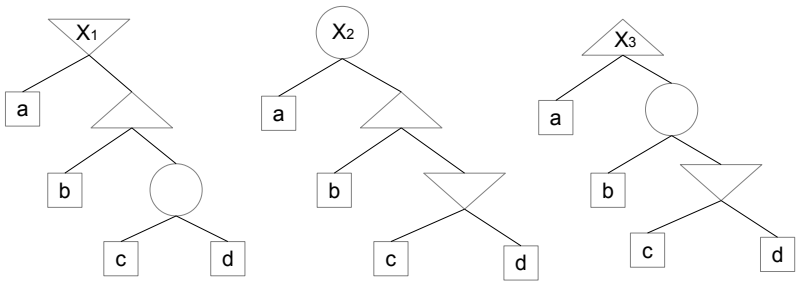


(d) Let X_1, X_2, and X_3 be the values at each root in the above minimax game trees. In these trees a, b, c, d, e, and f are constants (they are the same across all three trees). Determine which of the following statements are true for all possible assignments to constants a, b, c, d, d, and f.

(i) [1 pt] $X_1 = X_2$ True

(ii) [1 pt] $X_1 = X_3$ False

(iii) [1 pt] $X_2 = X_3$ False



(e) In this question we want to determine relations between the values at the root of the new game trees above (that is, between X_1, X_2, and X_3).

All three game trees use the same values at the leaves, represented by a, b, c, and d. The chance nodes can have any distribution over actions, that is, they can choose right or left with any probability. The chance node distributions can also vary between the trees.

For each case below, write the relationship between the values using $<$, $\leq$, $>$, $\geq$, $=$, or NR. Write NR if no relation can be confirmed given the current information. Briefly justify each answer (one sentence at most). (Hint: try combinations of $\{-\infty, -1, 0, 1, +\infty\}$ for a, b, c, and d.)

(i) [2 pts]

$X_1 \leq X_3$

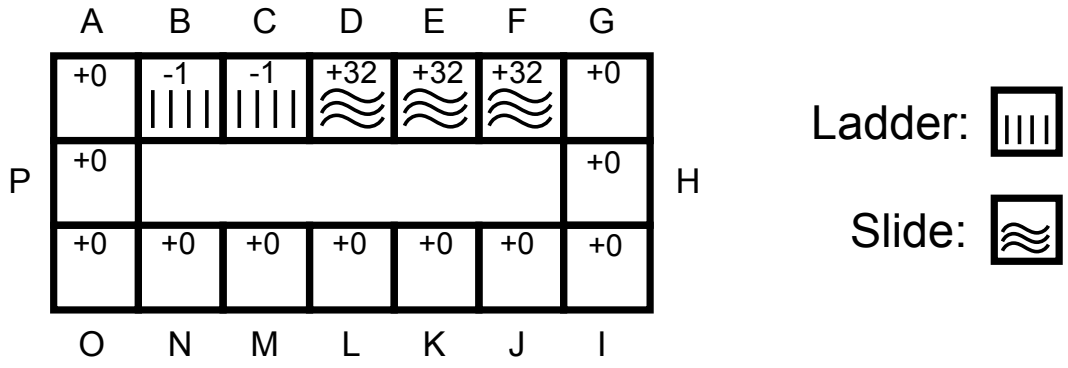
$X_1 \leq a$ and $X_3 \geq a$, thus $X_1 \leq X_3$. Equality is achieved by setting $a, b, c, d = 0$.

(ii) [2 pts]

X_2 NR X_3

NR, We can replace the min node by e. Let $a = -\infty, b = e = +\infty$ and the chance node always taking left, then $X_2 = -\infty \leq X_3 = +\infty$. Now, let $a = -\infty, b = +\infty, e = -\infty$ and the chance node always takes right, then $X_2 = +\infty > X_3 = -\infty$. Thus, there is not enough information to determine the relationship.

Q6. [7 pts] MDPs: Water Slide



Consider an MDP representing your experience at a water park, depicted by the figure above. Each state is labeled with a capital letter, A-P. The park has a single water slide which has a ladder that must be climbed (states B and C) before the slide can be ridden (states D, E, F). Apart from the slide states, you have three actions available in every state: stay where you are (Stay) or move to one of your two neighboring states (North, South, East, or West depending on the state's location). In the slides states (D, E, F) you only have one available action: East. Actions are deterministic: you always end up where you intend to.

Of course, you find it fun to ride the slide portion, but you hate exerting yourself while climbing the ladder. Walking around the rest of the park is a neutral experience. Specifically, you experience a reward of +32 when you enter a slide state, a reward of -1 upon entering a ladder state, and zero reward everywhere else. Note, you experience the reward of a state upon entering it (i.e. $R(s, a, s') = R(s')$). Let γ be the future reward discount.

(a) [1 pt] Suppose we run value iteration to convergence with $\gamma = 0.5$. Circle the action(s) from state A that are optimal under the calculated values.

South Stay **East**

(b) [2 pts] Suppose, instead, we run value iteration to convergence with $\gamma = 0.1$. Circle the action(s) from state A that are optimal under these calculated values.

South **Stay** East

(c) [1 pt] What happens to $V^*(A)$ as $\gamma \to 1$?

It goes to infinity.

(d) [1 pt] Suppose that $\gamma = 0.5$. How many iterations of value iteration must be done before the calculated value of state A is positive? In other words, what is the minimum n such that $V_n(A) > 0$?

3

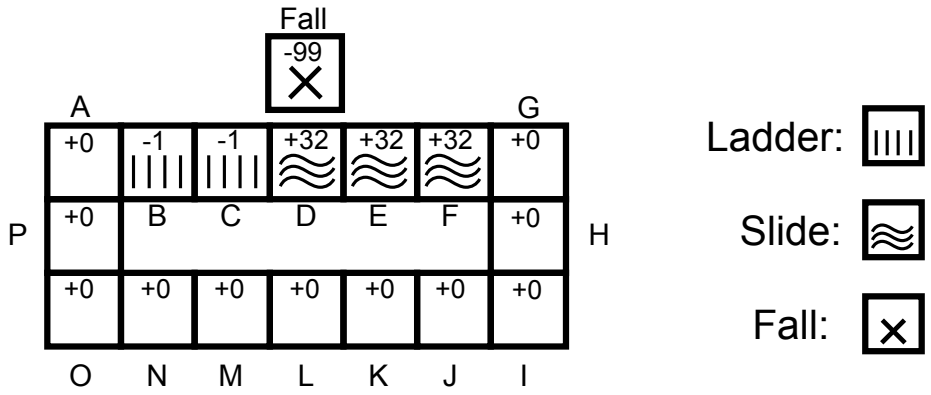
(e) [1 pt] Suppose that $\gamma = 0.5$. What is the calculated value of A after 13 iterations of value iteration? In other words, what is $V_{13}(A)$?

12.5

(f) [1 pt] Suppose that $\gamma = 0.5$. Let π be the policy that never leaves the current state unless that is the only available action. For example, under this policy, from state A you will always choose to remain in state A. What value for state A does policy evaluation converge to when you run it on this policy? In other words, what is $V^{\pi}(A)$?

0

Q7. [7 pts] RL: Dangerous Water Slide



Suppose now that several years have passed and the water park has not received adequate maintenance. It has become a dangerous water park! Now, each time you choose to move to (or remain at) one of the ladder or slide states (states B-F) there is a chance that, instead of ending up where you intended, you fall off the slide and hurt yourself. The cost of falling is -99 and results in you getting removed from the water park via ambulance. The new MDP is depicted above.

Unfortunately, you don't know how likely you are to fall if you choose to use the slide, and therefore you're not sure whether the fun of the ride outweighs the potential harm. You use reinforcement learning to figure it out!

For the rest of this problem assume $\gamma = 1.0$ (i.e. no future reward discounting). You will use the following two trajectories through the state space to perform your updates. Each trajectory is a sequence of samples, each with the following form: (s, a, s', r).

Trajectory 1: (A, East, B, -1), (B, East, C, -1), (C, East, D, +32)
Trajectory 2: (A, East, B, -1), (B, East, Fall, -99)

(a) [3 pts] What are the values of states A, B, and C after performing temporal difference learning with a learning rate of $\alpha = 0.5$ using only Trajectory 1?

$V(A) =$ -0.5 $\quad$ $V(B) =$ -0.5 $\quad$ $V(C) =$ 16

(b) [3 pts] What are the values of states A, B, and C after performing temporal difference learning with a learning rate of $\alpha = 0.5$ using both Trajectory 1 and Trajectory 2?

$V(A) =$ -1.0 $\quad$ $V(B) =$ -49.75 $\quad$ $V(C) =$ 16

(c) [1 pt] What are the values of states/action pairs (A, South), (A, East), and (B, East) after performing Q-learning with a learning rate of $\alpha = 0.5$ using both Trajectory 1 and Trajectory 2?

$Q(A, \text{South}) =$ 0.0 $\quad$ $Q(A, \text{East}) =$ -0.75 $\quad$ $Q(B, \text{East}) =$ -49.75

Q8. [12 pts] RL: Amusement Park

After the disastrous waterslide experience you decide to go to an amusement park instead. In the previous questions the MDP was based on a single ride (a water slide). Here our MDP is about choosing a ride from a set of many rides.

You start off feeling well, getting positive rewards from rides, some larger than others. However, there is some chance of each ride making you sick. If you continue going on rides while sick there is some chance of becoming well again, but you don't enjoy the rides as much, receiving lower rewards (possibly negative).

You have never been to an amusement park before, so you don't know how much reward you will get from each ride (while well or sick). You also don't know how likely you are to get sick on each ride, or how likely you are to become well again. What you do know about the rides is:

Actions / Rides	Type	Wait	Speed
Big Dipper	Rollercoaster	Long	Fast
Wild Mouse	Rollercoaster	Short	Slow
Hair Raiser	Drop tower	Short	Fast
Moon Ranger	Pendulum	Short	Slow
Leave the Park	Leave	Short	Slow

We will formulate this as an MDP with two states, well and sick. Each ride corresponds to an action. The 'Leave the Park' action ends the current run through the MDP. Taking a ride will lead back to the same state with some probability or take you to the other state. We will use a feature based approximation to the Q-values, defined by the following four features and associated weights:

Features	Initial Weights
$f_0(state, action) = 1$ (this is a bias feature that is always 1)	$w_0 = 1$
$f_1(state, action) = \begin{cases} 1 & \text{if } action \text{ type is Rollercoaster} \\ 0 & \text{otherwise} \end{cases}$	$w_1 = 2$
$f_2(state, action) = \begin{cases} 1 & \text{if } action \text{ wait is Short} \\ 0 & \text{otherwise} \end{cases}$	$w_2 = 1$
$f_3(state, action) = \begin{cases} 1 & \text{if } action \text{ speed is Fast} \\ 0 & \text{otherwise} \end{cases}$	$w_3 = 0.5$

(a) [1 pt] Calculate Q(*'Well'*, *'Big Dipper'*):

$1 + 2 + 0 + 0.5 = 3.5$

(b) [3 pts] Apply a Q-learning update based on the sample (*'Well'*, *'Big Dipper'*, *'Sick'*, -10.5), using a learning rate of $\alpha = 0.5$ and discount of $\gamma = 0.5$. What are the new weights?

Difference $= -10.5 + 0.5 * max(4, 3.5, 2.5, 2.0, 2.0) - 3.5 = -12$
$w_0 = 1 - 6 * 1 = -5$
$w_1 = 2 - 6 * 1 = -4$
$w_2 = 1 - 6 * 0 = 1$
$w_3 = 0.5 - 6 * 1 = -5.5$

(c) [2 pts] Using our approximation, are the Q-values that involve the sick state the same or different from the corresponding Q-values that involve the well state? In other words, is Q('*Well*', action) = Q('*Sick*', action) for each possible action? Why / Why not? (in just one sentence)

Same
They are the same because we have no features that distinguish between the two states.

Now we will consider the exploration / exploitation tradeoff in this amusement park.

(d) [2 pts] Assume we have the original weights from the table on the previous page. What action will an ϵ-greedy approach choose from the well state? If multiple actions could be chosen, give each action and its probability.
With probability $(1 - \epsilon\frac{4}{5})$ we will choose the Wild Mouse. Each other action will be chosen with probability $\frac{\epsilon}{5}$

(e) When running Q-learning another approach to dealing with this tradeoff is using an exploration function:

$$f(u, n) = u + \frac{k}{n}$$

(i) [1 pt] How is this function used in the Q-learning equations? (a single sentence)
The update replaces the max over Q values with a max over this function (with Q and N as arguments)

What are each of the following? (a single sentence each)

(ii) [1 pt] u :
The utility, given by Q

(iii) [1 pt] n :
The number of times this state has been visited

(iv) [1 pt] k :
A constant, by adjusting it we can change how optimistic we are about states we haven't visited much.

The potluck is coming up and the staff haven't figured out what to bring yet! They've pooled their resources and determined that they can bring some subset of the following items.

1. Pho
2. Apricots
3. Frozen Yogurt
4. Fried Rice
5. Apple Pie
6. Animal Crackers

There are five people on the course staff: Taylor, Jonathan, Faraz, Brian, and Alvin. Each of them will only bring one item to the potluck.

i. If (F)araz brings the same item as someone else, it cannot be (B)rian.
ii. (A)lvin has pho-phobia so he won't bring Pho, but he'll be okay if someone else brings it.
iii. (B)rian is no longer allowed near a stove, so he can only bring items 2, 3, or 6.
iv. (F)araz literally can't even; he won't bring items 2, 4, or 6.
v. (J)onathan was busy, so he didn't see the last third of the list. Therefore, he will only bring item 1, 2, 3, or 4.
vi. (T)aylor will only bring an item that is before an item that (J)onathan brings.
vii. (T)aylor is allergic to animal crackers, so he won't bring item 6. (If someone else brings it, he'll just stay away from that table.)
viii. (F)araz and (J)onathan will only bring items that have the same first letter (e.g. Frozen Yogurt and Fried Rice).
ix. (B)rian will only bring an item that is after an item that (A)lvin brings on the list.
x. (J)onathan and (T)aylor want to be unique; they won't bring the same item as anyone else.

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	

A		2	3	4	5	6
B		2	3			6
F	1		3		5	
J	1	2	3	4		
T	1	2	3	4	5	