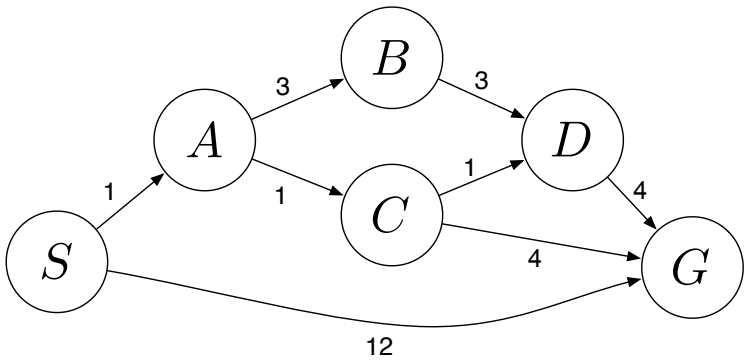
Table 1: For instructor's use

Question	Points Scored	Possible Points
1		12
2		8
3		12
4		12
5		12
6		8

This exam is closed book. You are allowed 2 sheets of notes (4 pages front and back). You may use any format for your notes that you like. Please explain all of your answers fully to receive full credit.

Here is some extra space. **Show all of your work on the questions!** If you need more paper just ask. Good luck!!

Question 1. Answer the following **graph search** questions using the graph below, where *S* is the start node and *G* is the goal node. Break any ties alphabetically.

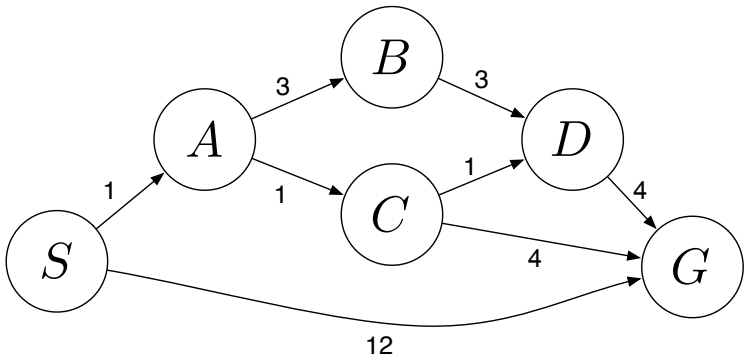


(a) (2 pts) What *solution path* would breadth-first graph search return for this problem?

(b) (2 pts) What *explored list and solution path* would uniform cost graph search return for this problem?

(c) (2 pts) What *explored list and solution path* would depth-first graph search return for this problem?

Graph repeated for your convenience:



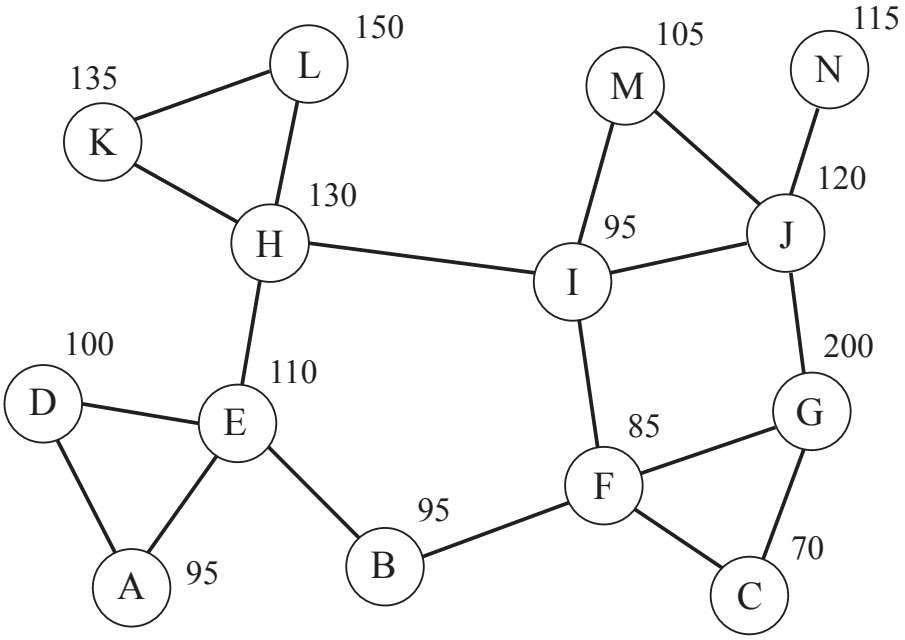
(d) (2 pts) What *solution path* would A* graph search, using a consistent heuristic, return for this problem?

(d) (4 pts) Design a consistent heuristic such that the explored list for A* graph search is:

Explored: S, A, B, C, G

Note that you can provide your answer by writing the heuristic value next to each state in the graph above.

Question 2. In this question you will use the **Hill Climbing search algorithm** on the graph shown below to *maximize the objective function*, which is given by the number next to each state.



(a) (1 pts) If hill climbing starts in state D, which adjacent state would be visited next?

(b) (3 pts) Give the order in which nodes would be visited by hill climbing, starting at D:

Visited: D_____

(c) (1 pts) Which state is the global maximum of the objective function?

(d) (3 pts) List all of the starting states with the property that hill climbing search will reach the global maximum.

Question 3. **Probabilistic Duck**

Your friend *Howard the Duck* has constructed a highly-biased probabilistic model of the animal kingdom, consisting of the joint probability distribution $P(W, Q, D)$, where W stands for "walks," Q stands for "quacks," and D stands for "is-a-duck."

	$W=0$		$W=1$	
	$Q=0$	$Q=1$	$Q=0$	$Q=1$
$D=0$	0.15	0.08	0.05	0.02
$D=1$	0.05	0.10	0.10	0.45

(a) (1 pts) Suppose that you are drawing random samples from this probabilistic model. What is the likelihood that a randomly-sampled animal will *walk*, *quack*, and *be a duck*?

(b) (4 pts) Calculate the following joint probability tables through marginalization (i.e. "summing out"): $P(W, Q)$, $P(D, Q)$, and $P(D)$. Note: we have provided the table $P(D, W)$ as an example.

	$W=0$	$W=1$
$D=0$	0.23	0.07
$D=1$	0.15	0.55

	$Q=0$	$Q=1$
$W=0$		
$W=1$		

	$Q=0$	$Q=1$
$D=0$		
$D=1$		

$D=0$	
$D=1$	

(c) (4 pts) If you are observing a duck, what is the probability that it can quack (i.e. $P(Q = 1|D = 1)$)? What is the probability that it can both quack and walk?

(d) (3 pts) You observe that a particular animal walks and quacks. What is the probability that it is a duck? Compare this answer to your answer in part (a). Are the probabilities the same or different? Explain why.

Question 4. Mark each of the following statements as *TRUE or FALSE*.

If FALSE, **rewrite the sentence** changing just a few words to make it true. Two points each.

- A game of poker (such as Texas Hold'em or Five Card Draw) is an example of a stochastic, fully-observable, multi-agent task environment.

- The primary difference between A* search and uniform cost search is the use of a priority queue.

- Given two admissable heuristics a and b, the heuristic defined by $\max\{a, b\}$ dominates both of them.

- The primary benefit of a reinforcement learning method like Monte Carlo Control over Policy Iteration is that Monte Carlo methods can learn directly from data, and therefore don't require a model.

- Given two random variables a and b, if $\sum_a p(a, b) = p(b)$, then we can conclude that b is independent of a.

- Given an MDP, suppose that we have run Value Iteration until convergence. If state c can be reached from state b by taking the optimal action a (e.g. $T(b, a, c) > 0$), then it follows that $V(c) \geq V(b)$.

Question 5. **Markov Decision Process**

Consider the following grid world for an MDP, where the states are $\{S_1, S_2, S_3\}$ and there are four goal terminals $\{G_1, G_2, G_3, G_4\}$. The reward for being in each state is -0.1. The reward for reaching G_1 is 10, while the reward for reaching each of the other goals is 1. The agent starts in S_1 and has *two* possible actions: *Right* and *Up*. The probabilistic transition model is as follows:

- When moving *Right*, the agent will go *Up* by accident with probability 0.2
- When moving *Up*, the agent will go *Right* by accident with probability 0.2.

1 G_2	1 G_3	1 G_4	
-0.1 S_1	-0.1 S_2	-0.1 S_3	10 G_1

(a) (6 pts) Assume that all of the *utilities are initialized to 0.1* and there is *no discounting* (i.e. $\gamma = 1$). Calculate the updated utilities for states S_1, S_2, and S_3 over *three rounds* of value iteration, by filling in the missing elements below. (Note that $U^i(S_j)$ denotes the utility for state j in iteration number i.)

$U^1(S_3) = \max_{R,U}\{$

$U^2(S_2) = \max_{R,U}\{$

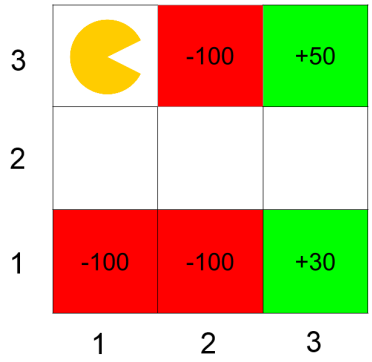
$U^3(S_1) = \max_{R,U}\{$

(b) (3 pts) When value iteration has converged, indicate on the grid world figure from the previous page what you believe the final (optimal) policy will be. How many iterations will be required for value iteration to converge (i.e. reach the point where the values no longer change)? Explain your answer.

(c) (3 pts) Demonstrate that your policy in part (b) is optimal using *policy iteration*. Policy iteration has converged when we compute the expected utility $U(\pi^*)$ for a policy π^*, and then demonstrate that π^* is optimal under $U(\pi^*)$.

Question 6. Q Learning

Consider the grid-world MDP below. Rewards are only awarded for taking the *Exit* action from one of the red or green (shaded) states. Taking this action moves the agent to the Done state, and the MDP terminates. Assume $\gamma = 1$ and $\alpha = 0.5$ for all calculations. If γ and α are needed in any equation then they must be included explicitly.



(a) (3 pts) The agent starts from the top left corner and you are given the following episodes from runs of the agent through this grid-world. Each line in an Episode is a tuple containing (s, a, s', r).

Episode 1	Episode 2	Episode 3	Episode 4	Episode 5
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), E, (3,2), 0	(2,2), S, (2,1), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
(3,2), N, (3,3), 0	(2,1), Exit, D, -100	(3,2), S, (3,1), 0	(3,2), N, (3,3), 0	(3,2), S, (3,1), 0
(3,3), Exit, D, +50		(3,1), Exit, D, +30	(3,3), Exit, D, +50	(3,1), Exit, D, +30

Fill in the following Q-values obtained from direct evaluation from the samples:

Q((3,2), N) = _____ Q((3,2), S) = _____ Q((2,2), E) = _____

(b) (3 pts) Q-learning is an online algorithm to learn optimal Q-values in an MDP with unknown rewards and transition function. The update equation is:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'))$$

where γ is the discount factor, α is the learning rate and the sequence of observations are $(\cdots, s_t, a_t, s_{t+1}, r_t, \cdots)$. Given the episodes in (a), fill in the time at which the following Q values first become non-zero. Your answer should be of the form (**episode#,iter#**) where **iter#** is the Q-learning update iteration in that episode. If the specified Q value never becomes non-zero, write *never*.

Q((1,2), E) = _____ Q((2,2), E) = _____ Q((3,2), S) = _____

(c) (2 pts) In Q-learning, we look at a window of $(s_t, a_t, r_{t+1}, s_{t+1})$ to update our Q-values. One can think of using an update rule that uses a larger window to update these values. Give an update rule for $Q(s_t, a_t)$ given the window $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, r_{t+2}, s_{t+2})$.

$Q(s_t, a_t) =$