1. (9 points) **True or False**

   Circle the correct answer for each T/F question. No need to explain the reasoning.

   (a) (1 point) True / False - A* Tree Search requires a consistent heuristic for optimality.

   (b) (1 point) True / False - Minimax is optimal against perfect opponents.

   (c) (1 point) True / False - There exist problems for which an admissible heuristic cannot be found.

   (d) (1 point) True / False - Uniform cost search with costs of 1 for all transitions is the same as depth first search.

   (e) (1 point) True / False - Alpha-Beta pruning can introduce errors during mini-max search.

   (f) (1 point) True / False - Each state can only appear once in a state graph.

   (g) (1 point) True / False - Policy Iteration always finds the optimal policy, when run to convergence.

   (h) (1 point) True / False - Higher values for the discount (γ) will, in general, cause value iteration to converge more slowly.

   (i) (1 point) True / False - For MDPs, adapting the policy to depend on the previous state, in addition to the current state, can lead to higher expected reward.

2. (10 points) **Short Answer** These short answer questions can be answered with a few sentences each.

   (a) (2 points) Briefly describe the relationship between admissible and consistent heuristics. When would you use each, and why?

   (b) (2 points) Briefly describe when you would use Alpha-beta pruning in minimax search.

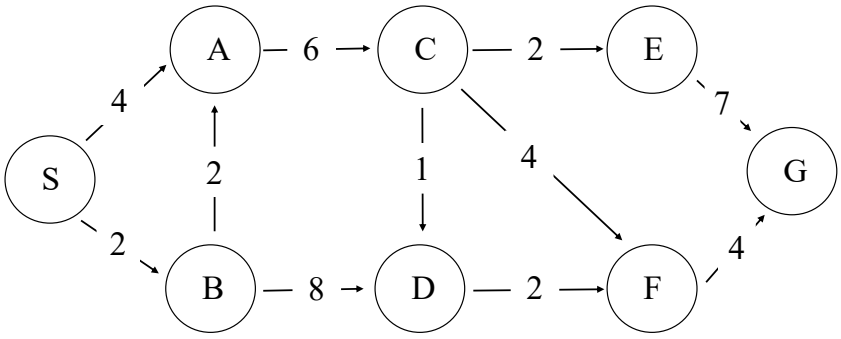
   (c) (2 points) Explain the relation between tree and graph search and when you would choose one over the other.

   (d) (2 points) Briefly describe the difference between UCS and A* search. When would you prefer to use each, and why?

   (e) (2 points) Describe a simple problem that breaks the Markov assumption of MDPs.

3. (10 points) **Informed Search**

You are given a graph below, and two heuristics functions h_1 and h_2. The task requires to start from state S and arrive at state G. Break any ties alphabetically (e.g., if two nodes are enqueued at the same time or have the same priority, first deque the node that has the lowest value alphabetically).



State	h_1	h_2
S	12	15
A	11	12
B	11	13
C	6	7
D	4	6
E	5	5
F	3	4
G	0	0

Now answer the following questions:

(a) (1 point) What is the cost of the optimal path for uniform cost search from S to G?

(b) (2 points) If using A^* search, is h_1 admissible? If not, provide a state at which the heuristic is inadmissible. Is it consistent? If not, provide an arc at which the heuristic is inconsistent.

(c) (2 points) If using A^* search, is h_2 admissible? If not, provide a state at which the heuristic is inadmissible. Is it consistent? If not, provide an arc at which the heuristic is inconsistent.

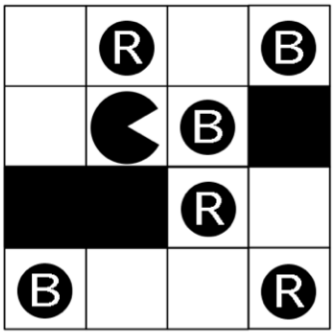
(d) (3 points) List all the visited nodes in the order of visiting them during (1) uniform cost search, (2) greedy search using h_2, and (3) A^* search using h_2. List in the format of $S \rightarrow \cdots \rightarrow G$.

(e) (2 points) For any given graph, is the path returned by greedy search always more expensive than the path returned by A* search? If you answer yes, explain; If you answer no, provide a simple counterexample. (Assume the heuristic used for A* is consistent and admissible.)

4. (7 points) **Pacman Search**

Consider a new Pacman game where there are two kinds of food pellets, each with a different color (red and blue). Pacman has peculiar eating habits; he strongly prefers to eat all of the red dots before eating any of the blue ones. If Pacman eats a blue pellet while a red one remains, he will incur a cost of 100 . Otherwise, as before, there is a cost of 1 for each step and the goal is to eat all the dots. There are K red pellets and K blue pellets, and the dimensions of the board are N by M.

In this particular game below, we have $K = 3, N = 4, M = 4$. Answer the following questions for any K, M, N in general.



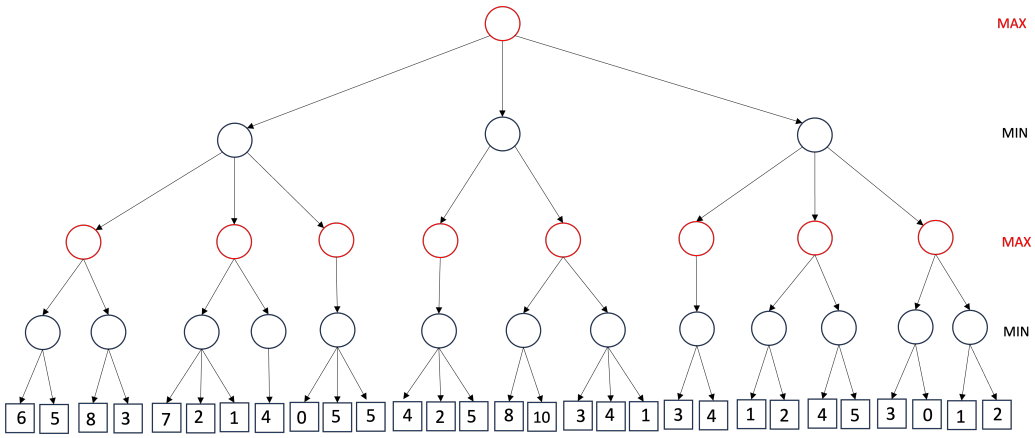
(a) (2 points) Name a search algorithm pacman could execute to get the optimal path? Briefly justify your choice (describe in one or two sentences)

(b) (5 points) Give a non-trivial admissible heuristic for this problem.

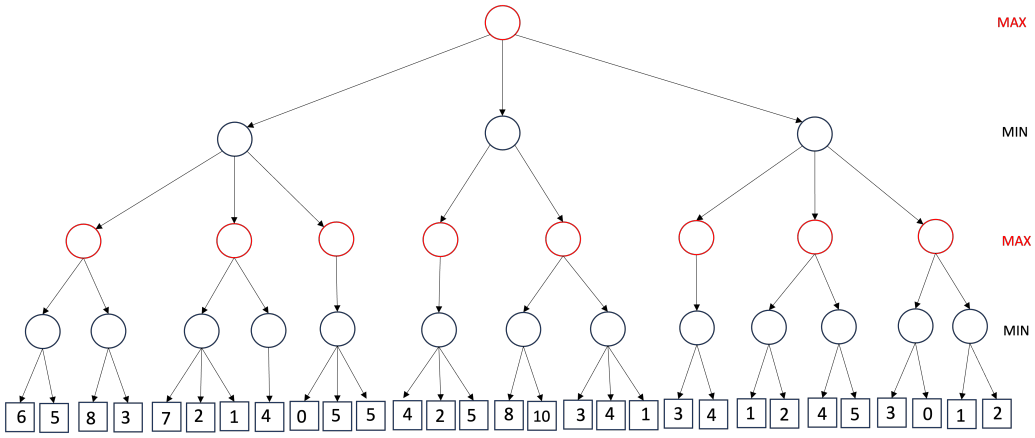
5. (10 points) **Minimax Search and Alpha-Beta Pruning**

Consider the depicted 2-player game tree below, with two optimal adversarial agents, the values at the leaf nodes are the terminal utilities for those states. Answer the following questions.

(a) (3 points) Write out or show the utility values for all intermediate nodes and root node using the Minimax algorithm.



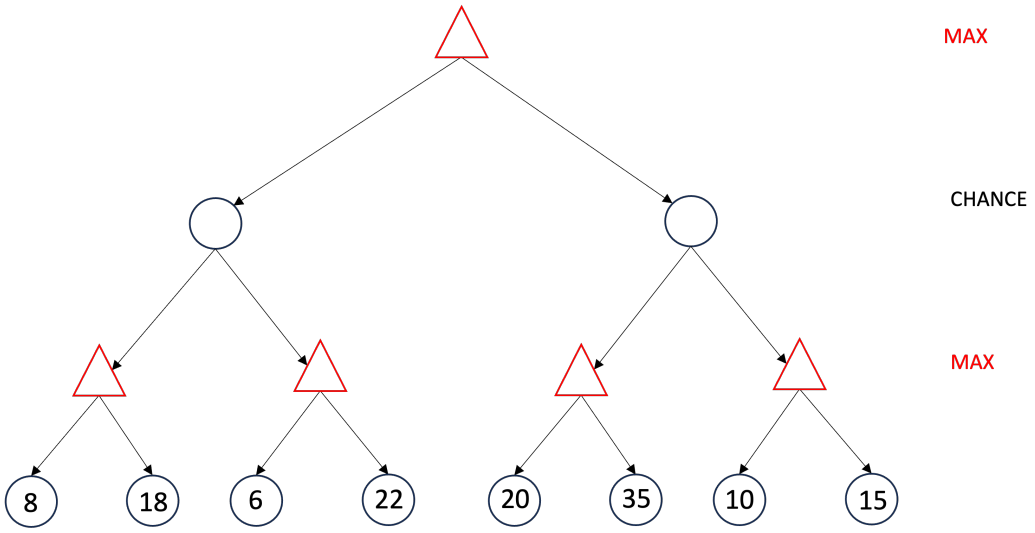
(b) (6 points) Using Alpha-Beta pruning, determine which branches are cut/unexplored, show your workings at all nodes, and use a **X** across a branch to indicate cutting a branch.



(c) (1 point) In a non zero-sum game (eg, Expectimax), would alpha-beta prune any nodes? if so why, and if not explain the reason as well?

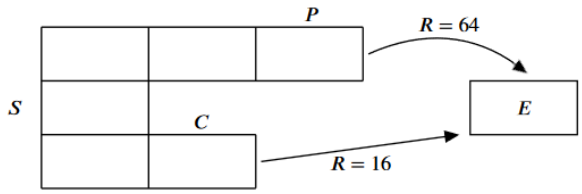
6. (5 points) **ExpectiMax**

   (a) (2 points) In the game tree below with a uniformly random adversarial opponent, Write out or show the utility values for all intermediate nodes and root node Using the Expecti-Max algorithm.



   (b) (3 points) Suppose after playing the game a few rounds, we observe that the opponent picks to go RIGHT at every branch with a probability of $1 - p$, calculate the range of values of p that would change the optimal path/decision taken at the root node, relative to the path/decision taken with uniform probabilities as done above.

7. (9 points) **Pacman College MDP** Pacman is now a CS student at UW. He finds himself in a (very) simplified, deterministic grid world MDP representation of UW depicted below, starting at state S. Pacman can take actions up, down, left or right. If an action moves him into a wall, he will stay in the same state. States C and P represent the CS building and the Party building respectively (their labels both appear above the relevant grid square). Pacman will study at the CS building or he will party at the Party building. At states C and P Pacman can take the exit action to receive the indicated reward and enter the terminal state, E. Note $R(s, a, s') = 0$ otherwise. Once in the terminal state the game is over and no actions can be taken. Let the discount factor $\gamma = \frac{1}{2}$ for this problem, unless otherwise specified.



(a) (1 point) Indicate the optimal policy for Pacman's college experience in the UW MDP gridworld. You may either construct a table or draw the policy on the grid.

(b) (1 point) How many value iterations k will it take before $V_k(P) = V^*(P)$?

(c) (1 point) Indicate all values which $V_k(P)$ will take on during the entire value iteration process.

(d) (1 point) How many value iterations k will it take before $V_k(S) = V^*(S)$?

(e) (1 point) Indicate all values which $V_k(S)$ will take on during the entire value iteration process.

We will now mess with Pacman's ability to party! You will be tasked with changing some of the MDP parameters so that Pacman no longer desires to visit the Party building and would instead prefer to study.

(f) (2 points) Provide a inequality bound on the discount factor γ that would make Pacman prefer to study over partying.

(g) (2 points) Tweak the reward function such that Pacman will always choose studying over partying. Write an inequality bound on $R(C, exit, E)$, that guarantees Pacman exits from C instead of P, leaving $R(P, exit, E)$ unchanged. Write N/A if this is not possible.