Name: **Student ID:**

CSE 573 Spring 2022 HW1

Due Friday 4/22 by 11:59pm

100 points

Instructions:

1) The homework should be done individually. Don't forget to write your name.
2) We highly recommend typing your homework, but writing and scanning, or annotating a PDF are also acceptable.
3) Keep your answers brief but provide enough explanations and details to let us know that you have understood the topic.
4) The assignment is due on April 22.
5) You should upload your assignments through gradescope.

Topics:	Points
Short Problems	15
Heuristics for Informed Search	20
Alpha-Beta pruning	15
Evaluation Function	20
Expectimax	30

Problem 1. Short Problems [15 Points]

A) Let's define the procedure of hill-climbing. You start at a random location on a hill, your goal is to get to the highest point on the hill. At each time step, you will take a step toward the location next to you that is higher than your current location.

   Is hill-climbing complete? Why? If not, is there any way to improve the performance in the discrete problem space? (3 points)

B) Pac-Man wants to get to the goal location from some initial position in a 2D grid.

   a. If Pac-Man wants to get to the goal location with the shortest path, what is the simplest state representation? Please use mathematical notations. (2 point)

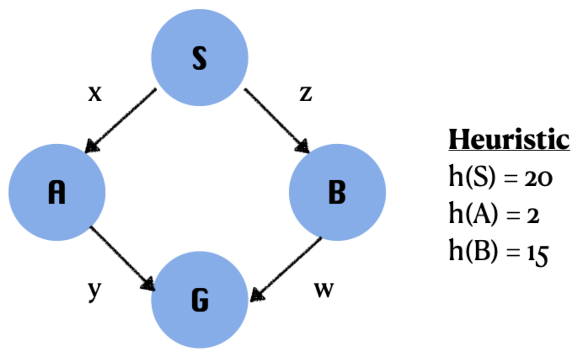
   b. If Pac-Man instead wants to first get to all four corners, then go to the goal location, should the state representation above change? If so, how? (2 point)

C) Search algorithm comparisons:

   In what circumstances is Greedy Search preferred over Uniform Cost search? Write down two circumstances. (2 points)

D) You are given the graph shown below, and the heuristics functions *h*. Your start from State S, and your goal is to go to State G.

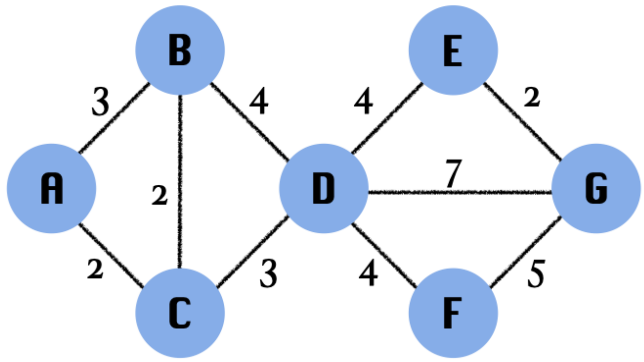
Find non-negative edge weights, x, y, z, and w, such that it satisfies each of the scenarios:



a. Scenario 1: (3 points)
Both greedy search and A* find the optimal solution.
x:
y:
z:
w:

b. Scenario 2: (3 points)
A* search finds the optimal solution, but greedy search doesn't.
x:
y:
z:
w:

Problem 2. Heuristics for Informed Search [20 Points]



Heuristic

Node	h1	h2
A	12.5	11
B	12	10
C	11	9
D	5	6
E	1	2
F	4	4.5
G	0	0

Consider the state space graph shown above. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions. Please refer to the search algorithms exactly **as presented on the lecture slides** as the ordering of the actions matters.

A) For each of the following graph search strategies, mark with an X which (if any) of the listed paths it could return. Note that for some search strategies the specific path returned might depend on tie-breaking behavior. In any such cases, make sure to mark **all paths** that could be returned under some tie-breaking scheme. (13 points)

Algorithm	**A-B-D-G**	**A-C-D-G**	**A-C-D-E-G**
BFS			
DFS			
UCS			
Greedy with heuristic h_1			
Greedy with heuristic h_2			
A* with heuristic h_1			
A* with heuristic h_2			

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

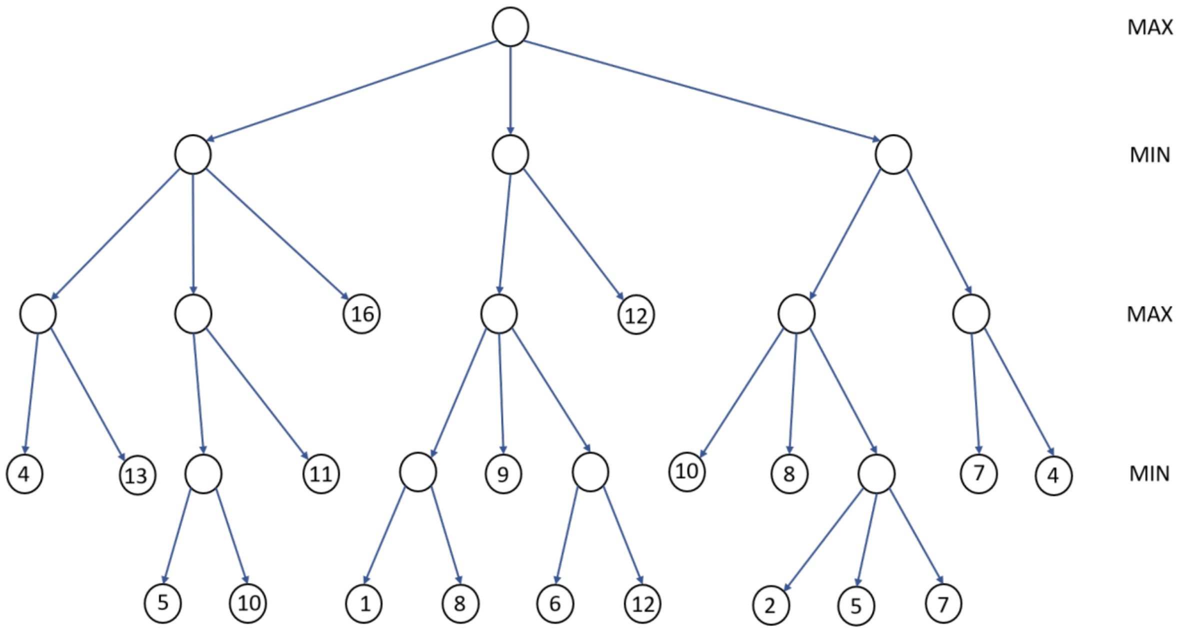
C) Is h_1 admissible? Is it consistent? Write an answer as well as the reason for each. (2 points)

D) Is h_2 admissible? Is it consistent? Write an answer as well as the reason for each. (2 points)

E) 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.) (2 points)

Problem 3. Alpha-Beta pruning [15 points]

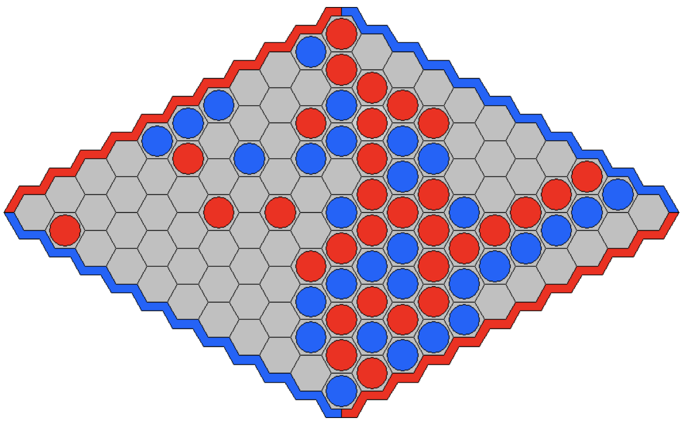
Below is the tree showing the states in a 2-player game played by two rational agents. This tree shows the 2-level expansion of decisions, and the values at the leaves are the utility values at those states.



A) Show the values of every intermediate node after performing the minimax algorithm. (5 points)
B) Use the Alpha-Beta pruning algorithm to determine the branches that need to be cut. (8 points)
C) For a general game tree (i.e., not limited to the above tree), are there any cases that the AlphaBeta algorithm gives a different value at the root node than the Minimax algorithm? If yes, show an example; if no, just say no. (2 points)

Problem 4. Evaluation Function [20 Points]

The Game of Hex was invented by John Nash in the 1940s. Hex consists of a rhombus game map divided into $n * n$ hexagons. Each player in a 2-player game has a marker (Blue and red). At each round a player can place a marker on an unmarked hexagon, and players alternate turns. The goal for the players is to link their opposite sides of the board in an unbroken chain. Whoever connects their sides first wins and receives +1 point and the opponent receives -1. It has been proven that a draw is impossible in Hex, hence there's always a winner.



A) Players can play optimally using a minimax algorithm. Why is expanding the whole game tree not practical? What are the things that we need to consider when we design the evaluation function so we can evaluate different stages of this game? (7 points)
B) Define an evaluation function that approximates the value of each state. (7 points)
C) If the size of the game is $n * n$ and each time the agent considers the next three moves (agents' move, minimizers' response, agents' subsequent move). What is the Big-θ time cost of the initial action? (6 points)

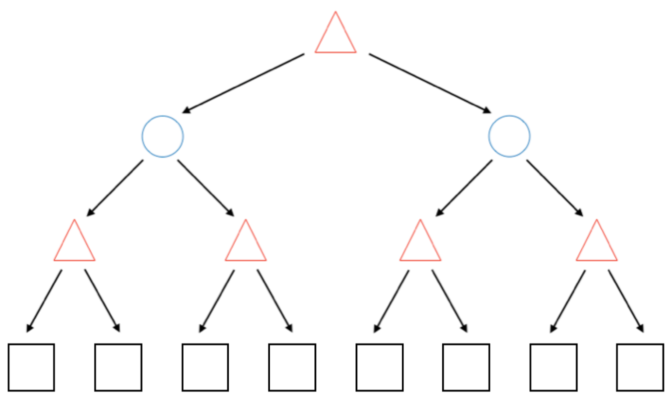
Problem 5 - Expectimax [30 points]

You have a friend visiting Seattle tomorrow. You want to make a dinner reservation for either patio or indoor seating. The weather forecast says it will rain with a 60% probability. Your friend's satisfaction will be as follows.

Weather	**Place**	**Your friend's satisfaction**
Rainy	Patio	0
Rainy	Indoor seating	50
Sunny	Patio	100
Sunny	Indoor seating	70

You can switch the reservation to the other seating option tomorrow, depending on the weather, but then your friend's satisfaction will be deducted by 20 due to a waiting time.

A) Imagine this is a game between you and the weather. Complete the following game tree by describing what each node/edge represents and filling the terminal values. (2 points)

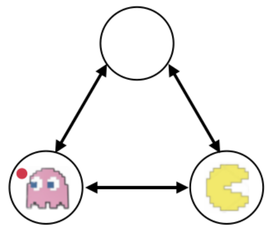


B) Fill in the values for interior nodes based on Expectimax search, and decide on whether you will reserve the patio or indoor seating. (5 points)

C) Suppose it will rain with probability *p* (instead of 60). Find the range of *p* that will change the optimal decision for the root max node. (5 Points)

D) Suppose it will rain with probability 60, but the deduction to the satisfaction if you change the reservation from indoor seating to patio is d (instead of 20). Find the range of d that will change the optimal decision for the root max node. (5 Points)

In the map below, the pacman (yellow) and the ghost (pink) can move to any adjacent circles by moving either clockwise or counterclockwise. Pacman starts first and alternates turns with the ghost. The game ends in either one of two cases: 1) pacman wins by eating the red dot, or 2) ghost wins by catching the pacman. When pacman meets the ghost and the red dot together, it will be caught by the ghost before eating the dot.



Pacman will receive the following scores

- Score of +*A* for eating a dot
- A penalty score of -2 if caught by the ghost.
- A penalty score of -1 every turn (either for pacman or ghost, including the turn pacman eats the dot or is eaten by the ghost)

Unfortunately, pacman does not know whether the movement of the ghost is random or adversarial.

E) Calculate the expected score of Pacman if it plays optimally either with Minimax or Expectimax, each in case of *A*=3 and *A*=100. Fill in the tables below. (8 points)

	Adversarial ghost	Random ghost
Minimax Pacman		
Expectimax Pacman		

[When *A*=3]

	Adversarial ghost	Random ghost
Minimax Pacman		
Expectimax Pacman		

[When *A*=100]

F) Suppose the ghost will be random with a probability of α and adversarial with a probability of $1 - \alpha$. Explain what pacman's movement should be at its first turn. You should use a function of *A* and α. (5 points)