Write a single line of code to create a list containing values from 1 to 100, [1, 2, ..., 100].

lst = (fill in the blank)

Q1.2
1 Point

Consider the following code. >>> mathematicians_list = ["Babbage", "Einstein", "Leibniz"] >>> mathematicians_list[:][0] = "Gödel"

Write the contents mathematicians_list after the above code has been executed and explain what causes the answer.

Q1.3
1 Point

Which of the following pieces of code will generate an error? Select all that apply.

A: {"a": 3, "b": 2, [1,2]: 2}
B: first_two = set(); first_two.add("one"); first_two.remove("two")
C: lst = [8, 9] + [2, 4]
D: {"A": 3, "B": 2} + {"C": 3, "D": 4}

Q1.4
1 Point

Which of the following pieces of code will generate an error? Select all that apply.

A: a = '123'; a[1] = '1'
B: b = (1, 2, 3); b[1] = 1
C: c = [1, 2, 3]; c[1] = 1
D: d = ({ }, { }); d[1][1] = 1

Q1.5
1 Point

In Python, what does the last command in this sequence print? >>> x = [(x, y) for x in [1,2,3] for y in [1,2,3]] >>> len(x)

A: 1
B: 3
C: 6
D: 9

Q2 Rational Agents & Environment
3 Points

Q2.1
1 Point

Which of the following meet the textbook's definition of an agent? (Check all that apply).

A: A pet hamster
B: A swimming pool
C: A thermostat
D: A bathroom faucet that automatically turns on

Q2.2

1 Point

An agent's choice of action at a given instant depends both on its percept sequence and future perceptual inputs.

A: True
B: False

Q2.3

1 Point

Randomized behavior can be rational in some environments.

A: True
B: False

Q3 Uninformed Search

6 Points

Q3.1

1 Point

What is the process of looking for a sequence of actions that reaches the goal called?

A: the agent function
B: the agent program
C: the search problem
D: the shortest path

Q3.2

1 Point

If there are possibly infinite paths in your search space, you should use:

A: BFS
B: DFS

Q3.3

1 Point

If you are working with restricted memory, it would be preferable to use:

A: BFS
B: DFS

Q3.4

1 Point

Which of the following are correct about the Uniform-cost search? (Check all that apply)

A: All moves are equal in path cost.
B: It is guaranteed to find an optimal solution.
C: Dijkstra's algorithm is a Uniform-cost Search without a goal.

Q3.5

1 Point

Depth-first search has, in general, much lower space complexity than iterative deepening.

A: True
B: False

Q3.6

1 Point

Depth-first search is complete.

A: True
B: False

Q4 Informed Search

8 Points

Q4.1

1 Point

Which of the following is true about greedy best-first search?

A: It always produces the most optimal solution when a solution is returned.
B: At each step, the node with the minimal value as computed by the heuristic function is expanded.
C: It expands nodes farthest from the goal first to reach the solution quickly.
D: It is complete and always returns a solution.

Q4.2

1 Point

Assume we have a heuristic function, *h1*, that has an effective branching factor (b*) of 5. Which of the following modifications would result in more efficient A* algorithm performance? (Check all that apply.)

A: Substituting an admissible heuristic function, *h2*, that has a b* value of 1.5.
B: Creating a new problem with more restrictions on the actions that can be taken.
C: Substituting an admissible heuristic function, *h3*, which dominates *h1*.
D: Substituting an admissible heuristic function, *h4*, such that *h4* (n) < *h1* (n) for any node n.

Q4.3

1 Point

Which of the following algorithms are extended from A* to reduce memory requirements and can use all available memory? (Check all that apply)

A: IDA* (iterative-deepening A*)
B: RBFS (Recursive best-first search)
C: MA* (memory-bounded A*)
D: SMA* (Simplified MA*)

Q4.4

1 Point

4.5 In estimating the travel distance between two locations, Euclidean distance is a consistent and admissible heuristic.

A: True
B: False

Q4.5

1 Point

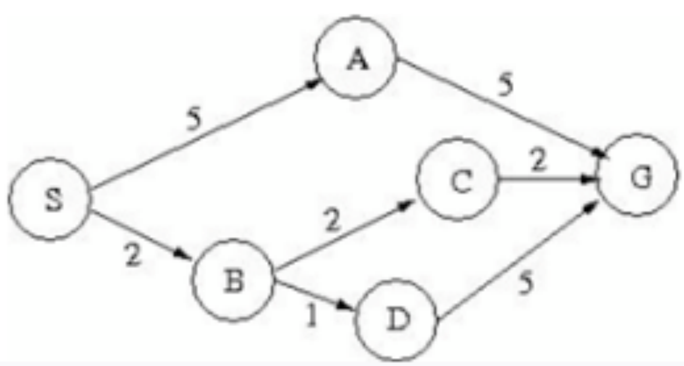
A* search uses heuristics to prune the search space so that the use of space is effectively O(bd).

A: True
B: False

Q4.6

3 Points

Consider the following search space where we want to find a path from the start state S to the goal state G. h3. The table shows three different heuristic functions h1, h2, and h3.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

4.7.1 What solution path is found by Greedy Best-first search using h2?

A: S, B, D, G
B: S, B, C, G

C: S, A, G
D: None of the above

4.7.2 What solution path is found by A* search using h1?

A: S, B, D, G
B: S, B, C, G
C: S, A, G
D: None of the above

4.7.3 What solution path is found by A* search using h3?

A: S, B, D, G
B: S, B, C, G
C: S, A, G
D: None of the above

Q5 5. Adversarial Search

4 Points

Q5.1

1 Point

You're playing a game where you roll a fair die once. You receive $100 multiplied by the value of the die. What is the expected monetary value (EMV) of playing this game?

Q5.2

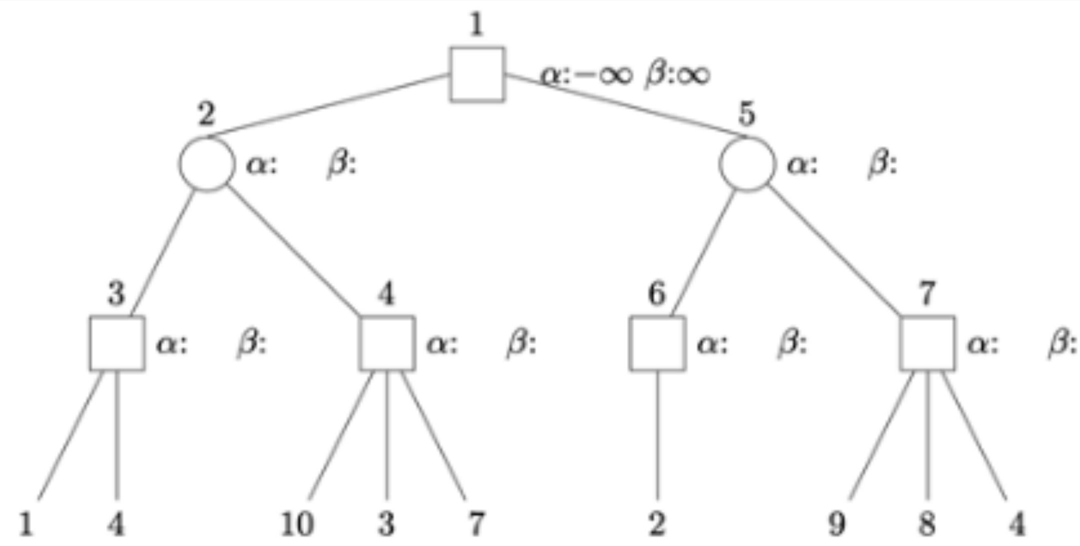
1 Point

For [____] games, chance nodes denoting the probabilities associated with each action as well as expecti-minimax values are included in the minimax tree. Please fill in the blank.

Q5.3

2 Points

The following tree represents all possible outcomes of a hypothetical zero-sum game:



This tree is from the perspective of the MAX player; MAX nodes are represented by squares and MIN nodes by circles. The leaves of the tree represent the value of the game for the MAX player. The number of each node indicates the order in which they are considered by the Minimax and α-β pruning algorithms.

5.3.1 What are the "back-up" values of each node in the tree using the Mini-max strategy? (The lists are ordered by the node numbers, 1 to 7.)

A: 3,3,1,3,4,2,4
B: 4,4,4,10,2,2,9
C: 10,10,4,10,9,2,9
D: 1,1,1,3, 2, 2, 4

5.3.2 Run the α-β pruning algorithm and list each leaf and node that would NOT be considered by the α-β pruning algorithm. (Assume that leaves are considered in left-to-right order.)

A: Nodes: 7; Leaves: 3, 7, 9, 8 ,4
B: Nodes: 5,6,7; Leaves: 2, 9, 8, 4
C: Nodes: 4, 7; Leaves: 10, 3, 7, 9, 8, 4
D: Nodes: None; Leaves: None

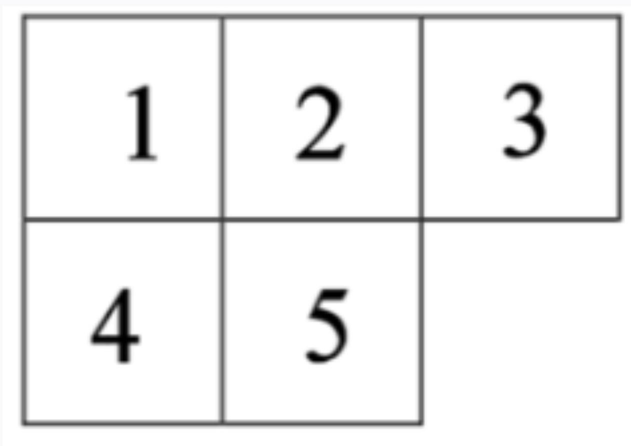
5.3.3 What is the α-β value of node 5 after running the α-β pruning algorithm?

A: -∞, + ∞
B: -∞, 4
C: 4, + ∞
D: 2, + ∞

Q6 CSP

6 Points

Consider the problem of assigning colors to the five squares on board below such that horizontally adjacent and vertically adjacent squares do not have the same color. Assume there are possible two colors, red (R) and black (B). Formulated as a constraint satisfaction problem, there are five variables (the squares) and two possible values (R, B) for each variable.



Q6.1

2.5 Points

If initially every variable has both possible values and we then assign variable 1 to have value R, what is the result of the Forward Checking algorithm?

Please rite down the set of possible values for each node. (sample answer 1 = {R, B}, etc.)

1 =

2 =

3 =

4 =

5 =

Q6.2

1 Point

If initially every variable has both possible values and AC-3 is run, what are the resulting domains for each of the variables?

Q6.3

2.5 Points

If initially every variable has both possible values except variable 5 has only value B, what is the result of *AC-3*?

1 =

2 =

3 =

4 =

5 =

Q7 Logical Agents

2 Points

Q7.1

1 Point

Which of the following terms appears in the CNF of (A ∧ B) ⇔ (C ∨ D) ?

A: A ∨ B ∨ ¬ C ∨ ¬ D
B: ¬ C ∨ B
C: D ∨ ¬ A
D: A ∧ B

Q7.2

1 Point

Which of the following reduction rules is NOT used in the construction of the CNF of (A ∧ B) ⇔ (C ∨ D)?

A: Biconditional Elimination
B: Implication Elimination
C: DeMorgan
D: Distributivity of ∧ over ∨