复查测验提交: Homework 1

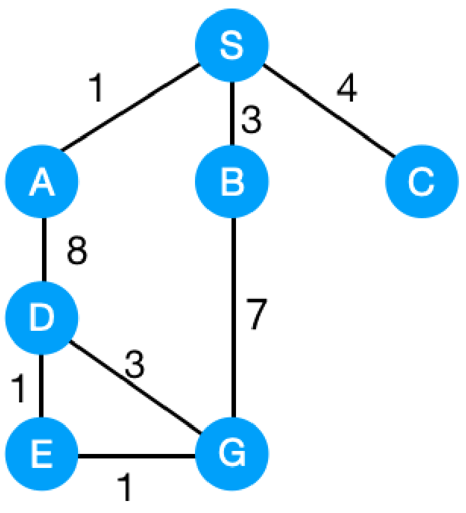
用户	信息科学与技术学院 周守琛
课程	人工智能I
测试	Homework 1
已开始	23-10-23 下午4:49
已提交	23-10-23 下午4:55
截止日期	23-10-25 下午11:59
状态	已完成
尝试分数	得 190 分，满分 190 分
已用时间	6 分钟
说明	注意：本作业不会自动提交。请在完成作业检查无误后，单击右下角“保存并提交”按钮提交作业。逾期未提交的作业不会被保存或计分。
显示的结果	所有答案, 已提交的答案, 正确答案

问题 1

得 10 分，满分 10 分

Graph Search Part 1

Consider a graph search from S to G on the graph below. Edges are labeled with action costs. Assume that ties are broken alphabetically (so a partial plan S->X->A would be expanded before S->X->B and S->A->Z would be expanded before S->B->A.)



Which search strategy will return the path S-B-G?

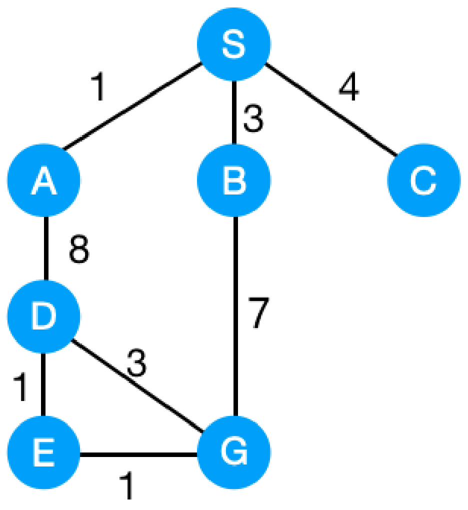
所选答案：
✓ Breadth-First Search
✓ Uniform Cost Search

答案：
✓ Breadth-First Search
Depth-First-Search
✓ Uniform Cost Search
None

问题 2

得 10 分，满分 10 分

Graph Search Part 2



Continue with Graph Search Part 2, which search strategy will return the path S-A-D-E-G?

所选答案:	✓ Depth-First Search
答案:	Breadth-First Search
	✓ Depth-First Search
	Uniform Cost Search
	None

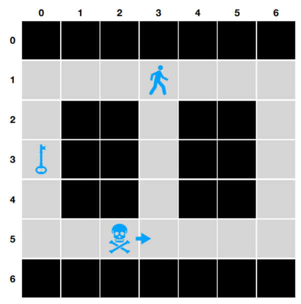
问题 3 得 5 分，满分 5 分

Uninformed Search and Heuristics Part 1

Consider the following simplified version of the classic Atari video game, *Montezuma's Revenge*: It is played on the board illustrated below. An agent (represented by the person icon in cell (1,3)) wishes to grab the key (in cell (3,0)). A skull starts in cell (5,2) and moves to the right by one cell after each action is executed until it ends up in the rightmost cell, at which point it starts moving to the left, and repeats this pattern back and forth.

The agent can be facing either left or right. There are 10 possible actions for the agent: 2 turning actions (*turn left*, *turn right*) and 8 moving actions (*left*, *right*, *up*, *down*, *left up*, *left down*, *right up*, *right down*). The agent can move up or down while facing either direction, but can move sideways or diagonally only if facing in that direction. For example, if the agent is facing right but tries to move *left up*, the agent will not move and nothing will happen. Furthermore, if the agent is already facing *left* and a *turn left* action is taken, nothing happens.

Lastly, the agent cannot move into a cell **currently occupied** by the skull, or a wall.



(a) Answer the following questions for the Montezuma's revenge board above:

Let N be the number of possible cell locations that the agent can be in, and let M be the number of possible cell locations that the skull can be in. Recall that for "pacman pathing", the representation of the state was (x, y) where x was the row and y was the column of pacman's position.

Which of the following is a minimal correct representation of a state in the state space for this game?

所选答案: ✓ A tuple (x, y) encoding the agent's x and y coordinates, agent's facing (*left*, *right*), t mod 12 where t is the time since game started.

答案: A tuple (x, y) encoding the agent's x and y coordinates, agent's facing (*left*, *right*), A tuple (x', y') encoding the skull's x and y coordinates, skull's facing (*left*, *right*).

✓ A tuple (x, y) encoding the agent's x and y coordinates, agent's facing (*left*, *right*), t mod 12 where t is the time since game started.

A tuple (x, y) encoding the agent's x and y coordinates, agent's facing (*left*, *right*).

A tuple (x, y) encoding the agent's x and y coordinates, agent's facing (*left*, *right*), y' encoding the skull's y coordinate, skull's facing (*left*, *right*).

A tuple (x, y) encoding the agent's x and y coordinates.

An integer d encoding the Manhatten distance from agent to key, agent's facing (*left*, *right*).

问题 4

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 2

What is the size of the state space in this question?

所选答案: ✓ $M \times N \times 4$

答案: $2^{M \times N+1}$

$2^{M \times N}$

✓ $M \times N \times 4$

$M \times N \times 2$

$M \times N$

$(M + N) \times 2$

None of the above

问题 5

得 5 分，满分 5 分

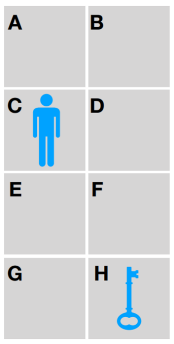
Uninformed Search and Heuristics Part 3

(b) Now, consider a simplified version of the board below, which has **no skull** and **no facing-direction for the agent** (i.e., the agent can move in any of the 8 directions as long as it remains in the board). For the three following graph search algorithms, perform the search procedure yourself and provide answers to the questions below regarding the nodes expanded during the search as well as the final path found by the algorithm.

On this board, assume that a diagonal move has a cost of 3, whereas moving left, right, up, or down has a cost of 1. Do notice the difference in costs, and recall which algorithms use this cost information and which algorithms do not.

Remember that the search procedure should begin at the agent's starting position (C). To break ties when adding nodes of equal cost to the fringe, follow alphabetical order.

Finally, when listing the order/number of nodes expanded, do not include nodes which are taken off the fringe but discarded immediately due to already having been visited.



(i) Breadth first graph search Fringe Data structure: queue Recall that BFS computes the smallest number of steps, $b(v)$, taken to get to a node v from the start node.

Order of nodes expanded (e.g. ABCD): **[1]**

Path returned (e.g. ABCD): **[2]**

Cost of path: **[3]**

1 的指定答案:	✓ CABDEFGH
2 的指定答案:	✓ CEH
3 的指定答案:	✓ 4

1 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	CABDEFGH	
✓ *完全匹配*	CABDEFG	

2 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	CEH	

3 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	4	

问题 6

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 4

What is $b(A), b(B), \cdots, b(H)$?

State s	A	B	C	D	E	F	G	H
$b(s)$	[4.1]	[4.2]	[4.3]	[4.4]	[4.5]	[4.6]	[4.7]	[4.8]

4.1 的指定答案:	✓ 1
4.2 的指定答案:	✓ 1
4.3 的指定答案:	✓ 0
4.4 的指定答案:	✓ 1
4.5 的指定答案:	✓ 1
4.6 的指定答案:	✓ 1
4.7 的指定答案:	✓ 2
4.8 的指定答案:	✓ 2

4.1 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.2 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.3 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	0	

4.4 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.5 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.6 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.7 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

4.8 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

问题 7

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 5

(ii)Uniform cost graph search Fringe data structure: priority queue (make sure you update/reorder the whole PQ after each addition) Recall that UCS keeps track of the lowest cost, $c(v)$, to get from the start node to the node v.

Order of nodes expanded (e.g. ABCD): **[1]**

Path returned (e.g. ABCD): **[2]**

Cost of path: **[3]**

1 的指定答案:	✓ CADEBFGH
2 的指定答案:	✓ CDFH
3 的指定答案:	✓ 3

1 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	CADEBFGH	
✓ *完全匹配*	CADEBFG	

2 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	CDFH	

3 的正确答案:		
评估方式	正确答案	区分大小写
✓ *完全匹配*	3	

问题 8

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 6

What is $c(A), c(B), \cdots, c(H)$?

State s	A	B	C	D	E	F	G	H
$c(s)$	**[4.1]**	**[4.2]**	**[4.3]**	**[4.4]**	**[4.5]**	**[4.6]**	**[4.7]**	**[4.8]**

4.1 的指定答案:	✓ 1
4.2 的指定答案:	✓ 2
4.3 的指定答案:	✓ 0
4.4 的指定答案:	✓ 1
4.5 的指定答案:	✓ 1
4.6 的指定答案:	✓ 2
4.7 的指定答案:	✓ 2
4.8 的指定答案:	✓ 3

4.1 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.2 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

4.3 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	0	

4.4 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.5 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	1	

4.6 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

4.7 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

4.8 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	3	

问题 9

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 7

(iii) *A graph search (with Manhattan distance to the goal as the heuristic)** Fringe data structure: priority queue (make sure you update/reorder the whole PQ after each addition) Recall that A* computes $f(v)$ for the nodes v that it expands, with $f(v) = c(v) + h(v)$ where $c(v)$ is the lowest cost to reach v from the start node and $h(v)$ is an estimate of the distance from v to the goal.

Order of nodes expanded during the search (e.g. ABCD): **[1]**

Path returned by the search (e.g. ABCD): **[2]**

Cost of path returned by the search: **[3]**

1 的指定答案:	✓ CDEFGH
2 的指定答案:	✓ CDFH
3 的指定答案:	✓ 3

1 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	CDEFGH	
✓ *完全匹配*	CDEFG	

2 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	CDFH	
3 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	

问题 10

得 5 分，满分 5 分

What is $f(A), f(B), \cdots, f(H)$? Note that here, we are asking for the true $f(v)$ values as dictated by the definition, which is the value populated by the search algorithm only if it were to expand every node. This particular search problem doesn't end up expanding all nodes, so the $f(v)$ estimate maintained by the algorithm on the queue is not the true $f(v)$ value that we're asking for. Hint: you can fill out these values directly by looking at the board.

State s	A	B	C	D	E	F	G	H
$f(s)$	[4.1]	[4.2]	[4.3]	[4.4]	[4.5]	[4.6]	[4.7]	[4.8]

4.1 的指定答案：	✓ 5
4.2 的指定答案：	✓ 5
4.3 的指定答案：	✓ 3
4.4 的指定答案：	✓ 3
4.5 的指定答案：	✓ 3
4.6 的指定答案：	✓ 3
4.7 的指定答案：	✓ 3
4.8 的指定答案：	✓ 3

4.1 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	5	
4.2 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	5	
4.3 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	
4.4 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	
4.5 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	
4.6 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	
4.7 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	
4.8 的正确答案：		
评估方式	**正确答案**	**区分大小写**
✓ *完全匹配*	3	

问题 11

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 9

(c) Given your answers above, which one finds the shortest path (in number of steps)? Which one finds the optimal path (in cost)?

BFS: **[1]**

UCS: **[2]**

A*: **[3]**

所选答案： **Uninformed Search and Heuristics Part 9**

(c) Given your answers above, which one finds the shortest path (in number of steps)? Which one finds the optimal path (in cost)?

BFS: ✓ **Shortest path**

UCS: ✓ **Optimal path**

A*: ✓ **Optimal path**

答案： **Uninformed Search and Heuristics Part 9**

(c) Given your answers above, which one finds the shortest path (in number of steps)? Which one finds the optimal path (in cost)?

BFS: ✓ **Shortest path**

UCS: ✓ **Optimal path**

A*: ✓ **Optimal path**

所有答案选项

- Please select an answer
- Shortest path
- Optimal path
- Neither shortest nor optimal

问题 12

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 10

(d) For the same board and setting as part (b), determine whether the following heuristics are admissible or consistent.

问题 | 正确匹配 | 所选匹配

State s	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Heuristic $h(s)$	4	4	3	2	1	0	0	0

正确匹配: ✓ D. Inadmissible and inconsistent
所选匹配: ✓ D. Inadmissible and inconsistent

State s	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Heuristic $h(s)$	4	3	2	2	1	1	0	0

正确匹配: ✓ B. Admissible but inconsistent
所选匹配: ✓ B. Admissible but inconsistent

State s	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Heuristic $h(s)$	3	3	3	2	1	0	0	0

正确匹配: ✓ B. Admissible but inconsistent
所选匹配: ✓ B. Admissible but inconsistent

State s	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Heuristic $h(s)$	0	0	0	0	0	0	0	0

正确匹配: ✓ A. Admissible and consistent
所选匹配: ✓ A. Admissible and consistent

所有答案选项

A. Admissible and consistent

B. Admissible but inconsistent

C. Inadmissible but consistent

D. Inadmissible and inconsistent

问题 13

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 11

(e) In the previous questions, perhaps you used "relaxed" heuristics; that is, you estimated the true cost of something by evaluating the cost for a simpler version of the problem (which is easier to compute). For example, using euclidean distance would be a "relaxed" heuristic to estimate the length of the shortest path.

Formally, we will define a **relaxed heuristic** as a function on a state that returns a value that is always admissible and consistent. In this problem, we will consider two changes ("skull" and "teleporation") to the board/game above, and we will reason about the effect of these changes on the consistency of heuristics.

(i) For this new version of the game, your friend Nancy suggests taking the old game setting from part (b) and now adding the ability for the agent to perform a maximum of 1 "teleportation" action during the game. That is, on one of the agent's moves, it can choose to jump from its current state to any other non-goal state on the board.

Nancy argues that in this new game, at least one previously consistent heuristic can become inconsistent. Is Nancy right?

Note: we define heuristics for this problem as being a function of only the cell location: They cannot incorporate anything that did not exist in the old version of the game that we are comparing to.

所选答案:	✓ 对
答案:	✓ 对
	错

问题 14

得 5 分，满分 5 分

Uninformed Search and Heuristics Part 12

(ii) For this new version of the board, your friend Ethan suggests adding the skull back to the old board setting from part (b), and having the skull move back and forth between the cells E and F.

Ethan argues that in this new board, at least one previously consistent heuristic can become inconsistent. Is Ethan right?

Note: we define heuristics for this problem as being a function of only the cell location: They cannot incorporate the location of the skull, since that did not exist in the old version of the board that we are comparing to.

所选答案:	✓ 错
答案:	对
	✓ 错

问题 15

得 10 分，满分 10 分

Memory Efficient Graph Search

Recall from lecture the general algorithm for GRAPH-SEARCH reproduced below.

```
function GRAPH-SEARCH(problem, fringe, strategy) return a solution, or failure
    closed ← an empty set
    fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
    loop do
        if fringe is empty then return failure
        node ← REMOVE-FRONT(fringe, strategy)
        if GOAL-TEST(problem, STATE[node]) then return node
        if STATE[node] is not in closed then
            add STATE[node] to closed
            for child-node in EXPAND(STATE[node], problem) do
                fringe ← INSERT(child-node, fringe)
            end
    end
```

Using GRAPH-SEARCH, when a node is expanded it is added to the closed set. This means that even if a node is added to the fringe multiple times it will not be expanded more than once. Consider an alternate version of GRAPH-SEARCH, MEMORY-EFFICIENT-GRAPH-SEARCH, which saves memory by (a) not adding node n to the fringe if STATE[n] is in the closed set, and (b) checking if there is already a node in the fringe with last state equal to STATE[n]. If so, rather than simply inserting, it checks whether the old node or the new node has the cheaper path and then accordingly leaves the fringe unchanged or replaces the old node by the new node.

By doing this the fringe needs less memory, however insertion becomes more computationally expensive.

More concretely, MEMORY-EFFICIENT-GRAPH-SEARCH is shown below with the changes highlighted.

```
function MEMORY-EFFICIENT-GRAPH-SEARCH(problem, fringe, strategy) return a solution, or failure
    closed ← an empty set
    fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
    loop do
        if fringe is empty then return failure
        node ← REMOVE-FRONT(fringe, strategy)
        if GOAL-TEST(problem, STATE[node]) then return node
        if STATE[node] is not in closed then
            add STATE[node] to closed
            for child-node in EXPAND(STATE[node], problem) do
                fringe ← SPECIAL-INSERT(child-node, fringe, closed)
            end
    end


function SPECIAL-INSERT(node, fringe, closed) return fringe
    if STATE[node] not in closed set then
        if STATE[node] is not in STATE[fringe] then
            fringe ← INSERT(node, fringe)
        else if STATE[node] has lower cost than cost of node in fringe reaching STATE[node] then
            fringe ← REPLACE(node, fringe)
```

Now, we've produced a more memory efficient graph search algorithm. However, in doing so, we might have affected some properties of the algorithm. Assume you run MEMORY-EFFICIENT-GRAPH-SEARCH with the A* node expansion strategy and a consistent heuristic, select all statements that are true.

所选答案: ✓ The EXPAND function can be called at most once for each state.
✓ The algorithm is complete.
✓ The algorithm will return an optimal solution.

答案: ✓ The EXPAND function can be called at most once for each state.
✓ The algorithm is complete.
✓ The algorithm will return an optimal solution.

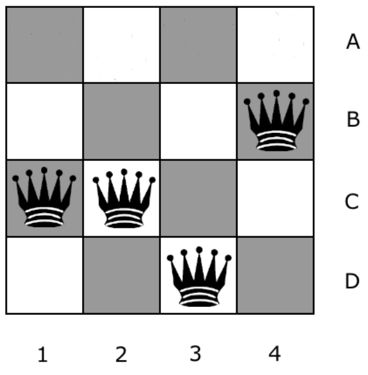
问题 16

得 5 分，满分 5 分

4-Queens Part 1

The min-conflicts algorithm attempts to solve CSPs iteratively. It starts by assigning some value to each of the variables, ignoring the constraints when doing so. Then, while at least one constraint is violated, it repeats the following: (1) randomly choose a variable that is currenly violating a constraint, (2) assign to it the value in its domain such that after the assignment the total number of constraints violated is minimized (among all possible selections of values in its domain).

In this question, you are asked to execute the min-conflicts algorithm on a simple problem: the 4-queens problem in the figure shown below. Each queen is dedicated to its own column (i.e. we have variables Q_1, Q_2, Q_3 and Q_4 and the domain for each one of them is $\{A, B, C, D\}$). In the configuration shown below, we have $Q_1 = C$, $Q_2 = C$, $Q_3 = D$, $Q_4 = B$. Two queens are in conflict if they share the same row, diagonal, or column (though in this setting, they can never share the same column).



You will execute min-conflicts for this problem three times, starting with the state shown in the figure above. When selecting a variable to reassign, min-conflicts chooses a conflicted variable at random. For this problem, assume that your random number generator always chooses the leftmost conflicted queen. When moving a queen, move it to the square in its column that leads to the fewest conflicts with other queens. If there are ties, choose the topmost square among them.

We recommend you work out the solutions to the following questions on a sheet of scratch paper, and then enter your results below.

(a) Starting with the queens in the configuration shown in the above figure, which queen will be moved, and where will it be moved to?

(a-1) Queen

所选答案：	✓ 1
答案：	✓ 1
	2
	3
	4

问题 17 得 5 分，满分 5 分

4-Queens Part 2

(a) Starting with the queens in the configuration shown in the above figure, which queen will be moved, and where will it be moved to?

(a-2) Position

所选答案：	✓ A
答案：	✓ A
	B
	C
	D

问题 18 得 5 分，满分 5 分

4-Queens Part 3

(b) Continuing off of Part 1-2, which queen will be moved, and where will it be moved to?

(b-1) Queen

所选答案：	✓ 2
答案：	1
	✓ 2
	3
	4

问题 19 得 5 分，满分 5 分

4-Queens Part 4

(b) Continuing off of Part 1-2, which queen will be moved, and where will it be moved to?

(b-2) Position

所选答案：	✓ A
答案：	✓ A
	B
	C
	D

问题 20 得 5 分，满分 5 分

4-Queens Part 5

(c) Continuing off of Part 3-4, which queen will be moved, and where will it be moved to?

(c-1) Queen

所选答案：	✓ 1
答案：	✓ 1
	2
	3
	4

问题 21 得 5 分，满分 5 分

4-Queens Part 6

(c) Continuing off of Part 3-4, which queen will be moved, and where will it be moved to?

(c-2) Position

所选答案：	✓ C
答案：	A
	B
	✓ C

D

问题 22

得 10 分，满分 10 分

Arc Consistency Part 1

Consider the problem of arranging the schedule for an event. There are three time slots: 1, 2, and 3. There are three presenters: A, B, and C. The variables for the CSP will then be A, B, and C, each with domain $\{1, 2, 3\}$. The following constraints need to be satisfied:

- A, B, and C all need to take on different values
- $A < C$

(a) Enforce consistency for the arc $A \to C$, and then select which values remain for each variable.

所选答案：
✓ $A:1$
✓ $A:2$
✓ $B:1$
✓ $B:2$
✓ $B:3$
✓ $C:1$
✓ $C:2$
✓ $C:3$

答案：
✓ $A:1$
✓ $A:2$
$A:3$
✓ $B:1$
✓ $B:2$
✓ $B:3$
✓ $C:1$
✓ $C:2$
✓ $C:3$

问题 23

得 10 分，满分 10 分

Arc Consistency Part 2

(b) Starting from the result of the previous step, enforce consistency for the arc $B \to A$, and then select which values remain for each variable.

所选答案：
✓ $A:1$
✓ $A:2$
✓ $B:1$
✓ $B:2$
✓ $B:3$
✓ $C:1$
✓ $C:2$
✓ $C:3$

答案：
✓ $A:1$
✓ $A:2$
$A:3$
✓ $B:1$
✓ $B:2$
✓ $B:3$
✓ $C:1$
✓ $C:2$
✓ $C:3$

问题 24

得 10 分，满分 10 分

Arc Consistency Part 3

(c) Starting from the result of the previous step, enforce consistency for the arc $C \rightarrow A$, and then select which values remain for each variable.

所选答案:
- ✓ $A : 1$
- ✓ $A : 2$
- ✓ $B : 1$
- ✓ $B : 2$
- ✓ $B : 3$
- ✓ $C : 2$
- ✓ $C : 3$

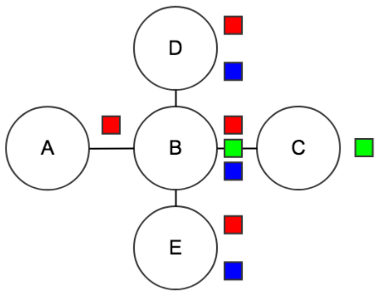
答案:
- ✓ $A : 1$
- ✓ $A : 2$
- $A : 3$
- ✓ $B : 1$
- ✓ $B : 2$
- ✓ $B : 3$
- $C : 1$
- ✓ $C : 2$
- ✓ $C : 3$

问题 25

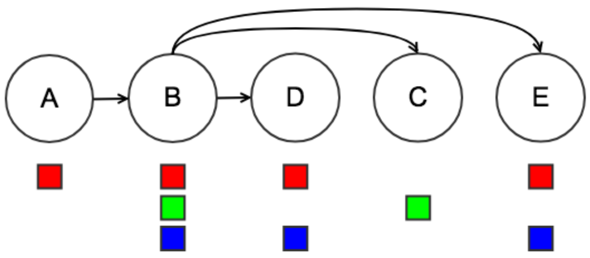
得 10 分，满分 10 分

Solving Tree-Structured CSPs Part 1

Consider the following tree-structured CSP that encodes a coloring problem in which neighboring nodes cannot have the same color. The domains of each node are shown.



The algorithm for solving tree-structured CSPs starts by picking a root variable. We can pick any variable for this. For this exercise, we will pick A. There are several linearizations consistent with A as the root; we will use the one shown below.



Step 1: Remove Backward

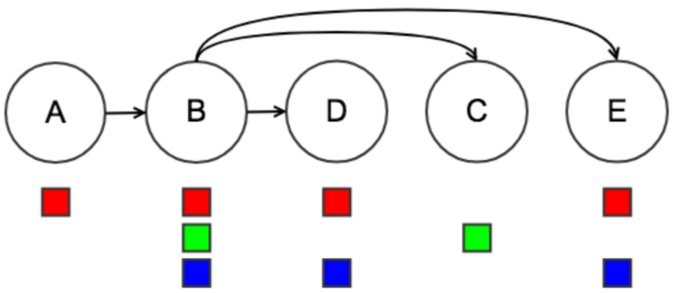
In this step we start with the right-most node (E), enforce arc-consistency for its parent (B),then do the same for the second-to-right-most node (C) and its parent (B), and so on. Execute this process, and then mark the remaining values for each variable below.

所选答案:
✓ A:red
✓ B:red
✓ B:blue
✓ C:green
✓ D:red
✓ D:blue
✓ E:red
✓ E:blue

答案:
✓ A:red
A:green
A:blue
✓ B:red
B:green
✓ B:blue
C:red
✓ C:green
C:blue
✓ D:red
D:green
✓ D:blue
✓ E:red
E:green
✓ E:blue

问题 26 得 10 分，满分 10 分

Solving Tree-Structured CSPs Part 2



Step 2: Assign Forward

Now that all domains have been pruned, we can find the solution in a single forward pass (i.e. no need for backtracking). This is done by starting at the left-most node (A), picking any value remaining in its domain, then going to the next variable (B), picking any value in its domain that is consistent with its parent, and continue left to right, always picking a value consistent with its parent's assignment.

If at any given node there are multiple colors left that are consistent with its parent's value, break ties by picking red over green, and then green over blue.

What is the solution found by running the algorithm?

所选答案: ✓ A:red
✓ B:blue
✓ C:green
✓ D:red
✓ E:red

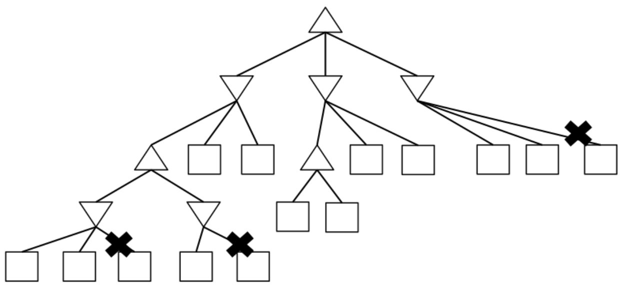
答案: ✓ A:red
A:green
A:blue
B:red
B:green
✓ B:blue
C:red
✓ C:green
C:blue
✓ D:red
D:green
D:blue
✓ E:red
E:green
E:blue

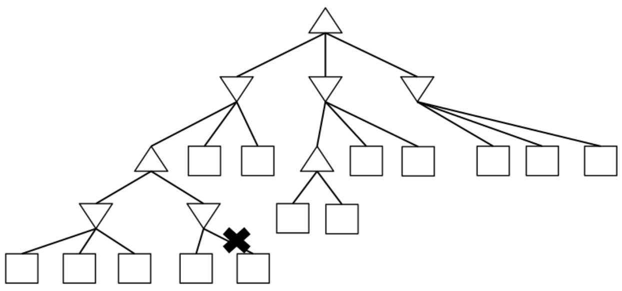
问题 27

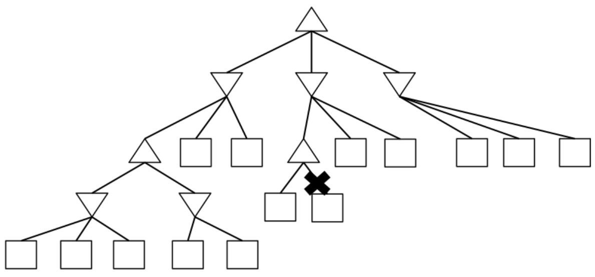
得 10 分，满分 10 分

Possible Pruning

Assume we run α-β pruning, expanding successors from left to right, on a game with trees as shown below.


(a)


(b)


(c)

所选答案: ✓ There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (b) will be achieved.

答案: There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (a) will be achieved.

✓ There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (b) will be achieved.

There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (c) will be achieved.

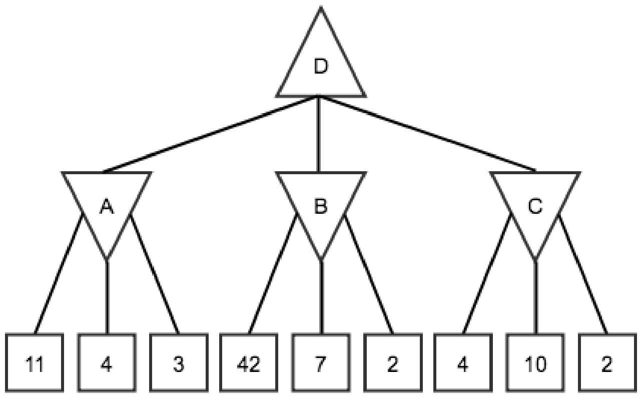
None of the above.

问题 28

得 10 分，满分 10 分

Minimax

Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Outcome values for the maximizing player are listed for each leaf node, represented by the values in squares at the bottom of the tree. Assuming both players act optimally, carry out the minimax search algorithm. Enter the values for the letter nodes in the boxes below the tree.



A**[X]**

B**[Y]**

C**[Z]**

D**[I]**

X 的指定答案:	✓ 3
Y 的指定答案:	✓ 2
Z 的指定答案:	✓ 2
I 的指定答案:	✓ 3

X 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	3	

Y 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

Z 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	2	

I 的正确答案:

评估方式	正确答案	区分大小写
✓ *完全匹配*	3	

2023年11月3日 星期五 下午12时33分52秒 CST

← **确定**