

# COMP 6721 Applied Artificial Intelligence (Summer 2023)

## Assignment #2: State Space Search

### Solutions

**Question 1** Once upon a time a farmer went to the market and purchased a fox, a goose, and a bag of beans. On his way home, the farmer came to the bank of a river and rented a boat. But in crossing the river by boat, the farmer could carry only himself and a single one of his purchases – the fox, the goose, or the bag of the beans.

If left alone, the fox would eat the goose, and the goose would eat the beans. The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact.

Represent this problem as a search problem. Choose a representation for the problem's states and:

(a) Write down the initial state

*Let  $position(farmer, fox, goose, beans)$  represent the position of the farmer, the fox, the goose and the beans with respect to the river. The possible positions are o for "original bank" and f for "far bank".*

*Initially, the state is:  $position(o, o, o, o)$*

(b) Write down the goal state

*$position(f, f, f, f)$*

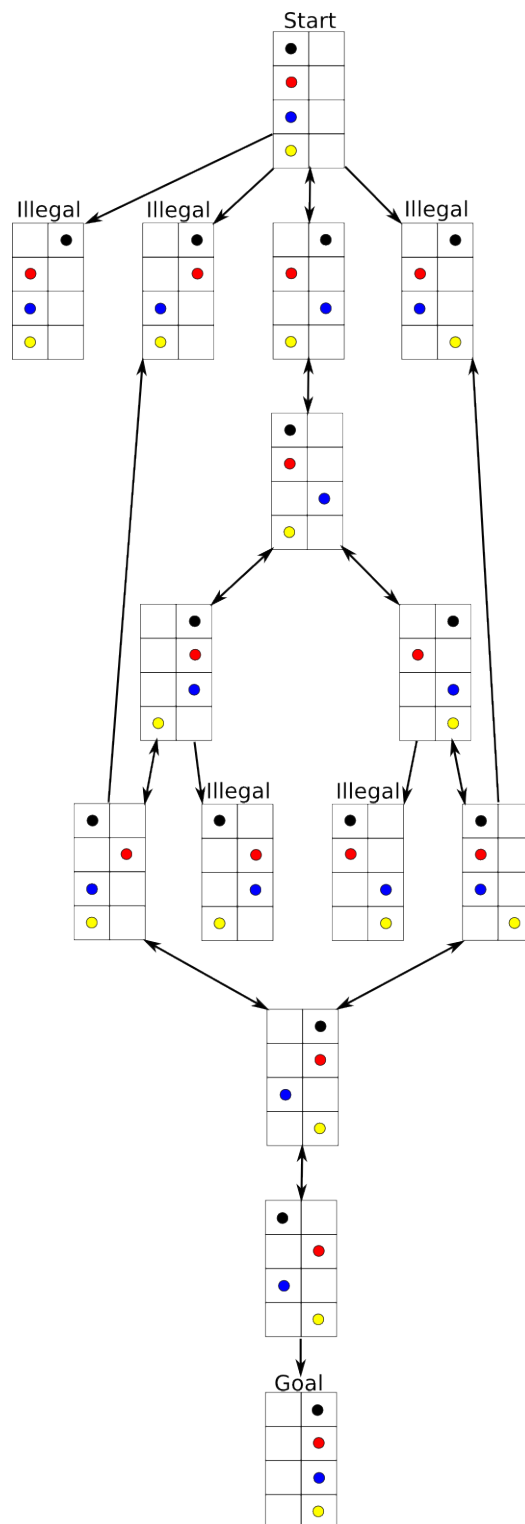
(c) Write down all illegal states

*$positions(f, o, o, o)$ ,  $positions(f, o, o, f)$ ,  $positions(o, f, f, o)$ ,  
 $positions(o, f, f, f)$ ,  $positions(f, f, o, o)$ ,  $positions(o, o, f, f)$ .*

(d) Write down the possible actions

*$moveFar(farmer, empty)$ ,  $moveFar(farmer, fox)$ ,  $moveFar(farmer, goose)$ ,  
 $moveFar(farmer, beans)$ ,  $moveBack(farmer, empty)$ ,  $moveBack(farmer, fox)$ ,  
 $moveBack(farmer, goose)$ ,  $moveBack(farmer, beans)$*

(e) Draw the state space for this problem



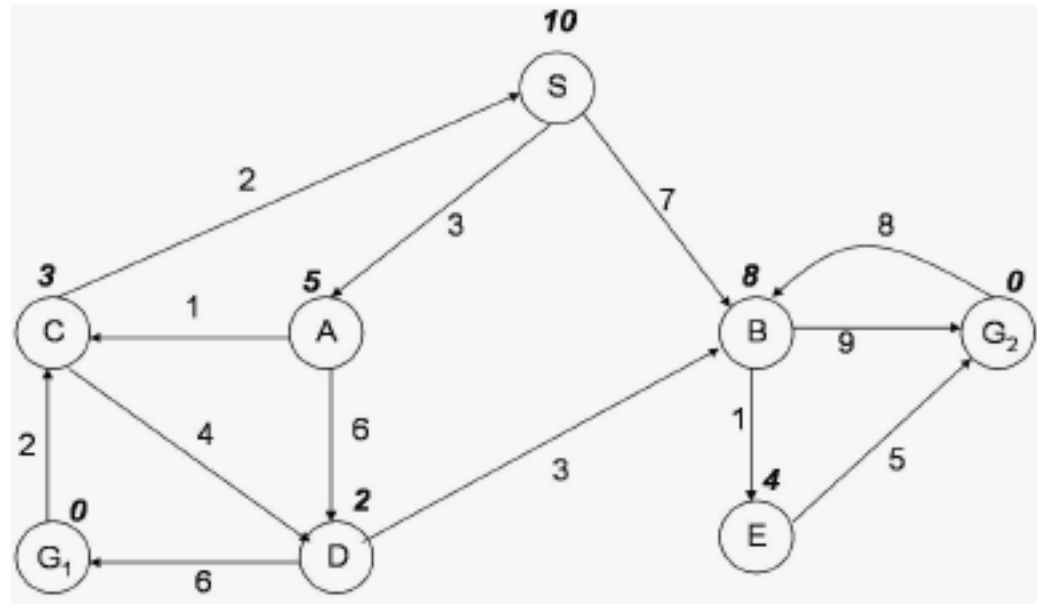
- (f) Find a series of moves to solve this problem

*moveFar(farmer, goose), moveBack(farmer, empty), moveFar(farmer, fox),  
moveBack(farmer, goose), moveFar(farmer, beans), moveBack(farmer, empty),  
moveFar(farmer, goose).*

*OR*

*moveFar(farmer, goose), moveBack(farmer, empty), moveFar(farmer, beans),  
moveBack(farmer, goose), moveFar(farmer, fox), moveBack(farmer, empty),  
moveFar(farmer, goose).*

**Question 2** Assume that  $S$  is the initial state and  $G_1$  and  $G_2$  are the goal states. The possible actions between states are indicated by arrows. The number labelling each arc is the actual cost of the action. For example, the cost of going from  $S$  to  $A$  is 3. The number in bold italic near each state is the value of the heuristic function  $h$  at that state. For example, the value of  $h$  at state  $C$  is 3. When all else is equal, expand states in alphabetical order.



For the following search strategies, show the states visited, along with the open and closed lists at each step (where it applies).

(a) Breadth-first search

Step	Visited State	Closed List	Open List
1			S
2	S	S	A, B
3	A	S, A	B, C, D
4	B	S, A, B	C, D, E, G2
5	C	S, A, B, C	D, E, G2
6	D	S, A, B, C, D	E, G2, G1
7	E	S, A, B, C, D, E	G2, G1
8	G2	S, A, B, C, D, E, G2	G1

(b) Depth-first search

Step	Visited State	Closed List	Open List
1			S
2	S	S	A, B
3	A	S, A	C, D, B
4	C	S, A, C	D, B
5	D	S, A, C, D	G1, B
6	G1	S, A, C, D, G1	B

(c) Iterative deepening depth-first search

*Note: Depth of each state in the open list is in parenthesis.*

*Depth 1:*

Step	Visited State	Closed List	Open List
1			S(1)
2	S	S	

*Depth 2:*

Step	Visited State	Closed List	Open List
1			S(1)
2	S	S	A(2), B(2)
3	A	S, A	B(2)
4	B	S, A, B	

*Depth 3:*

Step	Visited State	Closed List	Open List
1			S(1)
2	S	S	A(2), B(2)
3	A	S, A	C(3), D(3), B(2)
4	C	S, A, C	D(3), B(2)
5	D	S, A, C, D	B(2)
6	B	S, A, C, D, B	E(3), G2(3)
7	E	S, A, C, D, B, E	G2(3)
8	G2	S, A, C, D, B, E, G2	

(d) Uniform cost search

Step	Visited State	Closed List	Open List
1			S(0)
2	S	S	A(3), B(7)
3	A	S, A	C(4), B(7), D(9)
4	C	S, A, C	B(7), D(8); lower cost to D replaces previous cost
5	B	S, A, C, B	D(8), E(8), G2(16)
6	D	S, A, C, B, D	E(8), G1(14), G2(16)
7	E	S, A, C, B, D, E	G2(13), G1(14); G2(13) replaces G2(16)
8	G2	S, A, C, B, D, E, G2	

(e) Hill climbing

*Note: Hill climbing does not use open and closed lists, instead the first neighboring state with a better heuristic than the current state is visited.*

Step	Visited State	Evaluations
1	S	A(5) < S(10)
2	A	C(3) < A(5)
3	C	D(2) < C(3)
4	D	B(8) $\nless D(2)$ ; G1(0) < D(2)
5	G1	

(f) Best-first search

Step	Visited State	Closed List	Open List
1			S(10)
2	S	S	A(5), B(8)
3	A	S, A	D(2), C(3), B(8)
4	D	S, A, D	G1(0), C(3), B(8)
5	G1	S, A, D, G1	C(3), B(8)

(g) Algorithm A

Step	Visited State	Closed List	Open List
1			S(10)
2	S	S	A(3+5=8), B(7+8=15)
3	A	S, A	C(3+1+3=7), D(3+6+2=11), B(15)
4	C	S, A, C	D(3+1+4+2=10), B(15)
5	D	S, A, C, D	G1(3+1+4+6+0=14), B(15)
6	G1	S, A, C, D, G1	

**Question 3** *Exercise from OpenAI:*<sup>1</sup> Winter is here. You and your friends were tossing around a Frisbee at the park when you made a wild throw that left the Frisbee out in the middle of the lake. The water is mostly frozen, but there are a few holes where the ice has melted. If you step into one of those holes, you'll fall into the freezing water. At this time, there's an international Frisbee shortage, so it's absolutely imperative that you navigate across the lake and retrieve the disc as soon as you can. The surface is described using a rectangular grid like the figure below:

You are here			
	Hole		Hole
			Hole
Hole			Frisbee is here

- (a) What are the initial and goal state?

*Let a  $4 \times 4$  matrix represent the above grid. The position of each cell in the grid can then be addressed by the indices of the elements of the matrix. Hence we have:*

- *Initial state: (1,1)*
- *Goal state: (4,4)*

- (b) What are the illegal states?

*The illegal states are the holes in the grid with the following positions:*

*Illegal State 1: (2,2)*

*Illegal State 2: (2,4)*

*Illegal State 3: (3,4)*

*Illegal State 4: (4,1)*

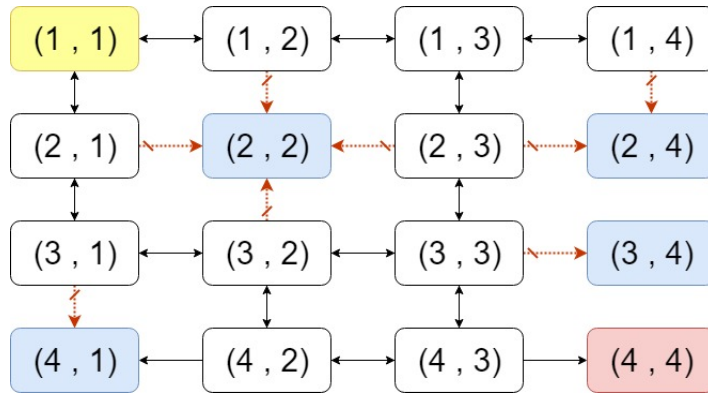
- (c) What are the possible actions and what should be their costs?

*The possible actions in a grid world are moving left, right, up, and down. Since we would like to reach the goal state as soon as possible (i.e. minimizing the number of actions), then we can assign a constant uniform cost for each action, for example a cost of 1. The algorithm aims to minimize the cost while reaching the goal state.*

- (d) Draw the state space for this problem.

*The state space is shown in the figure below, where the initial state is colored yellow, the goal state is colored red, and illegal states are colored blue.*

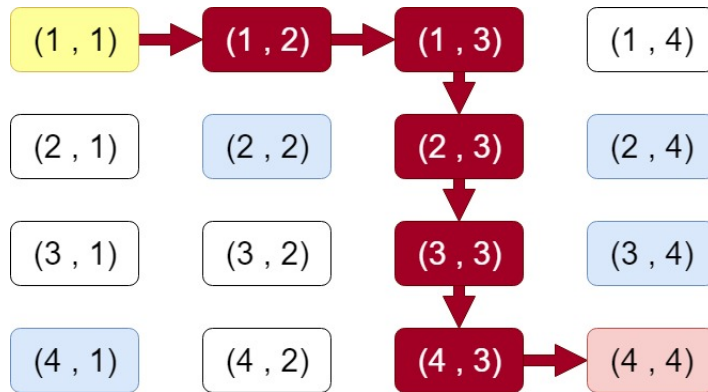
<sup>1</sup><https://gym.openai.com/envs/FrozenLake-v0/>



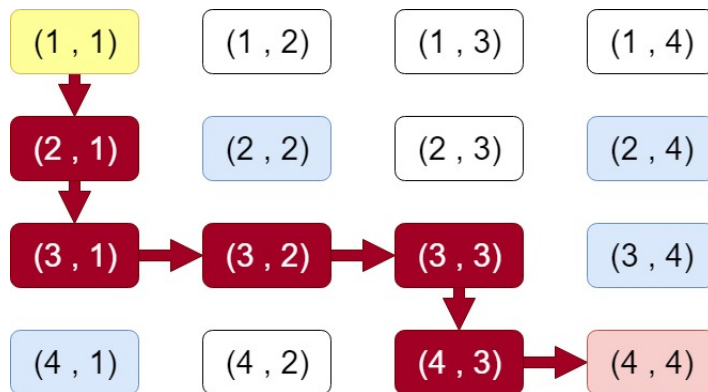
(e) Find a series of moves to solve this problem.

*Any of the following series of moves constitute a possible solution for this problem:*

*Path 1:*  $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (3, 3) \rightarrow (4, 3) \rightarrow (4, 4)$

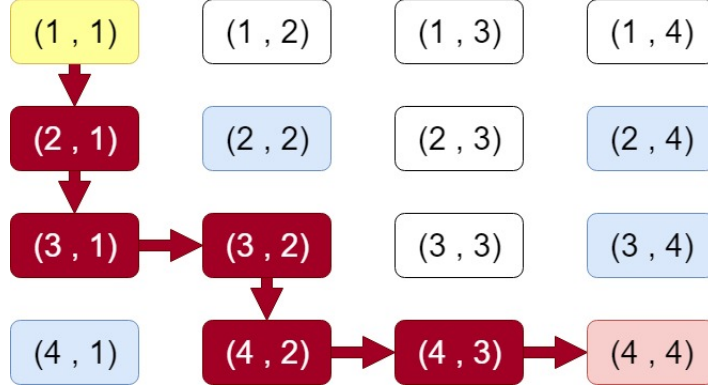


*Path 2:*  $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (3, 2) \rightarrow (3, 3) \rightarrow (4, 3) \rightarrow (4, 4)$





Path 3:  $(1, 1) \rightarrow (2, 1) \rightarrow (3, 1) \rightarrow (3, 2) \rightarrow (4, 2) \rightarrow (4, 3) \rightarrow (4, 4)$



- (f) Perform the following search strategies, show the states visited, along with the open and closed lists at each step.

I Breadth-first search

*The detailed execution of breadth-first search is listed below. A pictorial representation of the progress of this algorithm can be seen in Figure [1].*

Step 1 • Visited:

- open:  $\{(1, 1)\}$
- closed:  $\emptyset$

Step 2 • Visited:  $(1, 1)$

- open:  $\{(1, 2), (2, 1)\}$
- closed:  $\{(1, 1)\}$

Step 3 • Visited:  $(1, 2)$

- open:  $\{(2, 1), (1, 3)\}$
- closed:  $\{(1, 2), (1, 1)\}$

Step 4 • Visited:  $(2, 1)$

- open:  $\{(1, 3), (3, 1)\}$
- closed:  $\{(2, 1), (1, 2), (1, 1)\}$

Step 5 • Visited:  $(1, 3)$

- open:  $\{(3, 1), (1, 4), (2, 3)\}$
- closed:  $\{(1, 3), (2, 1), (1, 2), (1, 1)\}$

Step 6 • Visited:  $(3, 1)$

- open:  $\{(1, 4), (2, 3), (3, 2)\}$
- closed:  $\{(3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$

Step 7 • Visited:  $(1, 4)$

- open:  $\{(2, 3), (3, 2)\}$
- closed:  $\{(1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$

Step 8 • Visited:  $(2, 3)$

- *open*:  $\{(3, 2), (3, 3)\}$
- *closed*:  $\{(2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- Step 9* • *Visited*:  $(3, 2)$ 
  - *open*:  $\{(3, 3), (4, 2)\}$
  - *closed*:  $\{(3, 2), (2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- Step 10* • *Visited*:  $(3, 3)$ 
  - *open*:  $\{(4, 2), (4, 3)\}$
  - *closed*:  $\{(3, 3), (3, 2), (2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- Step 11* • *Visited*:  $(4, 2)$ 
  - *open*:  $\{(4, 3)\}$
  - *closed*:  $\{(4, 2), (3, 3), (3, 2), (2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- Step 12* • *Visited*:  $(4, 3)$ 
  - *open*:  $\{(4, 4)\}$
  - *closed*:  $\{(4, 3), (4, 2), (3, 3), (3, 2), (2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- Step 13* • *Visited*:  $(4, 4)$ 
  - *open*:  $\{\}$
  - *closed*:  $\{(4, 4), (4, 3), (4, 2), (3, 3), (3, 2), (2, 3), (1, 4), (3, 1), (1, 3), (2, 1), (1, 2), (1, 1)\}$
- *The goal state is observed at this point and the search has terminated.*

## II Uniform cost search

*Since we have assumed that all step cost are equal, then the uniform-cost search and breadth-first search will have the same result.*

## III Depth-first search

*The detailed execution of depth-first search is listed below. A pictorial representation of the progress of this algorithm can be seen in Figure [2].*

- Step 1* • *Visited*:
  - *open*:  $\{(1, 1)\}$
  - *closed*:  $\emptyset$
- Step 2* • *Visited*:  $(1, 1)$ 
  - *open*:  $\{(1, 2), (2, 1)\}$
  - *closed*:  $\{(1, 1)\}$
- Step 3* • *Visited*:  $(1, 2)$ 
  - *open*:  $\{(1, 3), (2, 1)\}$
  - *closed*:  $\{(1, 2), (1, 1)\}$

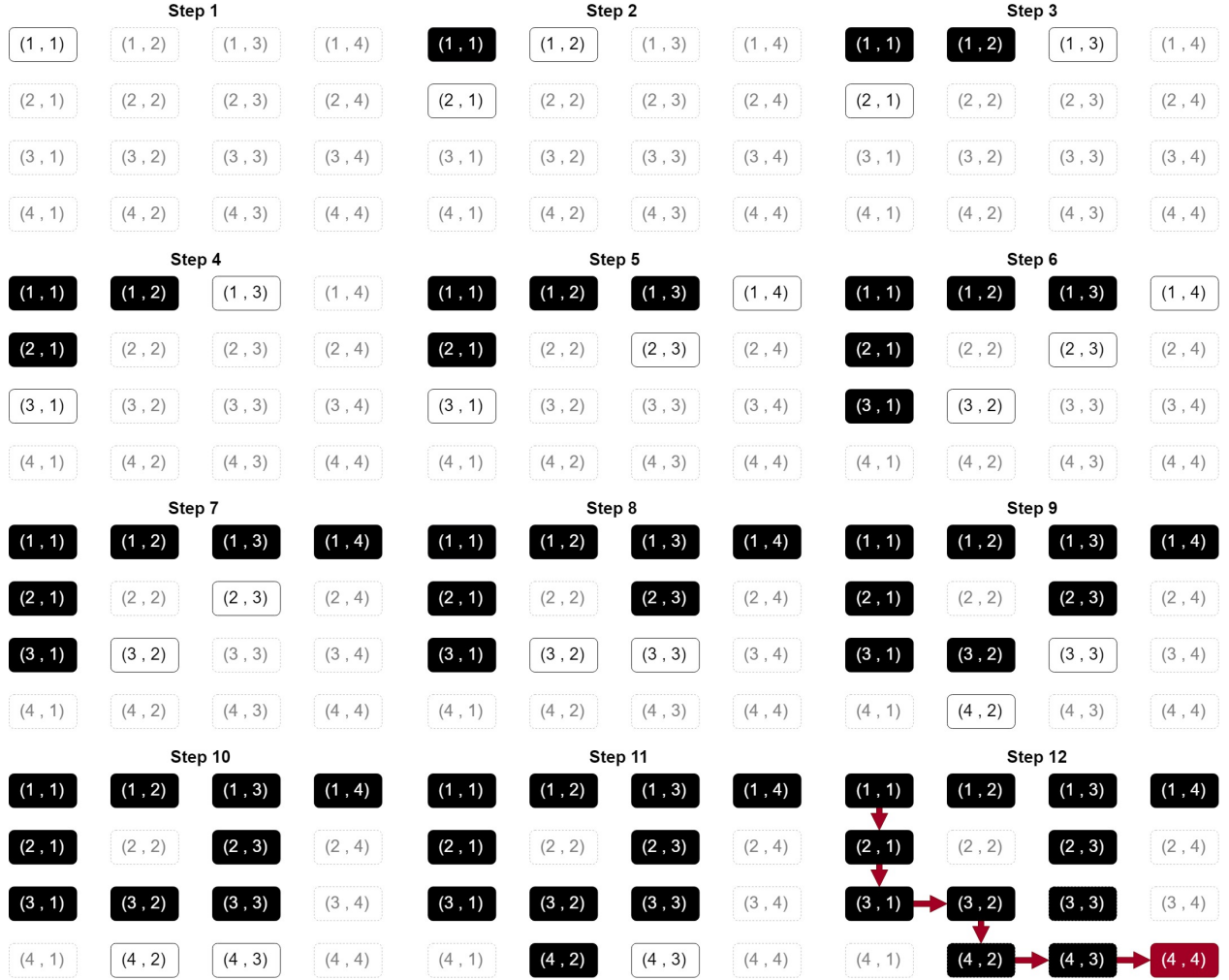


Figure 1: Progress of breadth-first search algorithm is depicted above. The cells with solid white background represent the open set. The cells with solid black background represent the closed set and the cell with solid red background is the goal state.

- Step 4* • *Visited:* (1, 3)
- *open:* {(1, 4), (2, 3), (2, 1)}
  - *closed:* {(1, 3), (1, 2), (1, 1)}
- Step 5* • *Visited:* (1, 4)
- *open:* {(2, 3), (2, 1)}
  - *closed:* {(1, 4), (1, 3), (1, 2), (1, 1)}
- Step 6* • *Visited:* (2, 3)
- *open:* {(3, 3), (2, 1)}
  - *closed:* {(2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 7* • *Visited:* (3, 3)
- *open:* {(3, 2), (4, 3), (2, 1)}
  - *closed:* {(3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 8* • *Visited:* (3, 2)
- *open:* {(3, 1), (4, 2), (4, 3), (2, 1)}
  - *closed:* {(3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 9* • *Visited:* (3, 1)
- *open:* {(4, 2), (4, 3), (2, 1)}
  - *closed:* {(3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 10* • *Visited:* (4, 2)
- *open:* {(4, 3), (2, 1)}
  - *closed:* {(4, 2), (3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 11* • *Visited:* (4, 3)
- *open:* {(4, 4), (2, 1)}
  - *closed:*  
{(4, 3), (4, 2), (3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
- Step 12* • *Visited:* (4, 4)
- *open:* {(2, 1)}
  - *closed:*  
{(4, 4), (4, 3), (4, 2), (3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)}
  - *The goal state is observed at this point and the search has terminated.*

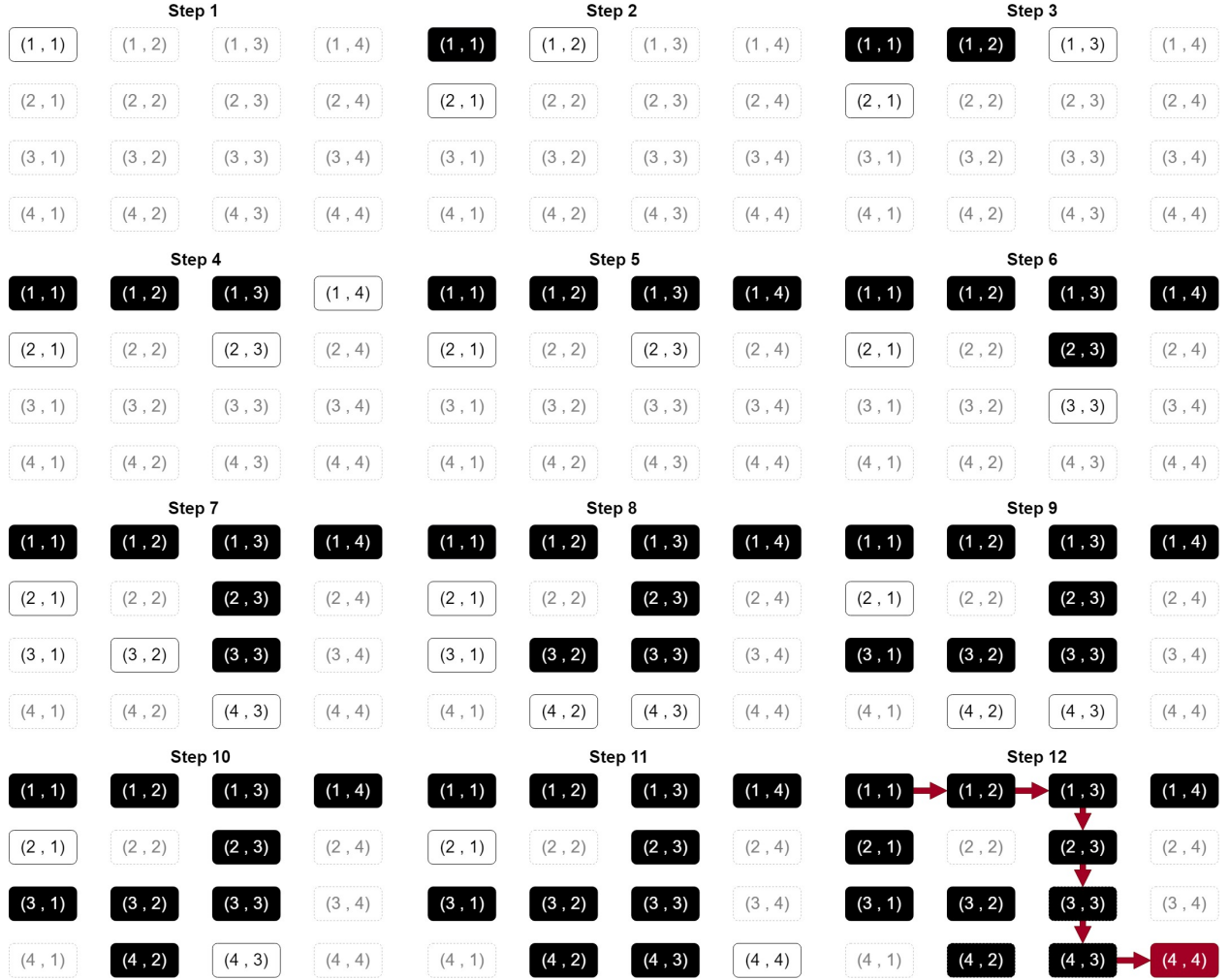


Figure 2: Progress of depth-first search algorithm is depicted above. The cells with solid white background represent the open set. The cells with solid black background represent the closed set and the cell with solid red background is the goal state.