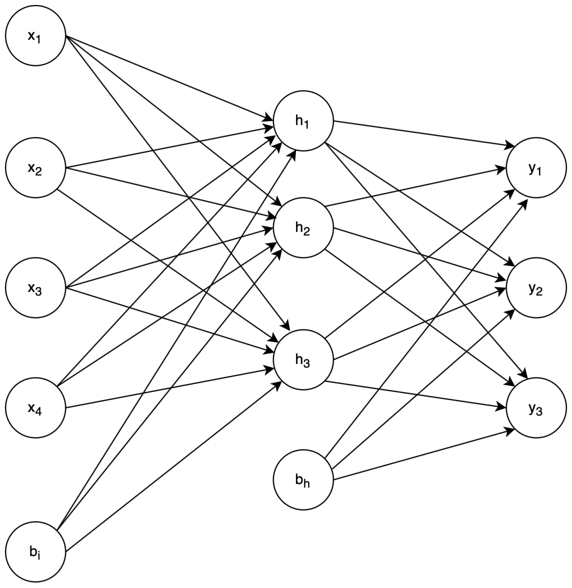
COMP 472: Artificial Intelligence
Artificial Neural Networks
Solutions

1 Question

Assume the following neural network with 4 input nodes, 1 hidden layer and 3 output layers.



Now suppose we have a binary classification task on a small dataset where each sample has five features. Does the above network suitable for this task? Explain your answer.

Solution

The answer is No! Since each input has five features, the network with four input nodes cannot be used. Note that in the above network, there are five nodes where the first four nodes are the input nodes and the last one is the input bias.

On the other hand, for a binary classification tasks we have two outputs; class one and class two. Hence two output nodes would be enough.

2 Question

Suppose the same neural network as the question above, with the input, bias, weights and outputs indicated below. Find the output values for an input x.

Assume that all the bias values are +1, and have the initial weight of 1. All other network weights are initialized to 0.5 and sigmoid is used as the activation function. The input values, initial values of hidden nodes, and the outputs are as follows:

$$x = [1, 0, 0.3, 0.7] \qquad w_{ih} = \begin{bmatrix} 0.5, 0.5, 0.5 \\ 0.5, 0.5, 0.5 \\ 0.5, 0.5, 0.5 \\ 0.5, 0.5, 0.5 \end{bmatrix} \qquad w_{ho} = \begin{bmatrix} 0.5, 0.5, 0.5 \\ 0.5, 0.5, 0.5 \\ 0.5, 0.5, 0.5 \end{bmatrix} \qquad y = [?, ?, ?]$$

Solution

To find out the output values, we have to multiply each values of x with the corresponding values of w_ih which is the weights from input nodes to the hidden layer. After that, the biases b_{ih} will be added and the sigmoid activation function will be applied on them.

weighted avg + bias

$$h_{in} = xw_{ih} + b_{ih} = [1., 1., 1.] + [1, 1, 1] = [2., 2., 2.]$$

$$h = \sigma(h_{in}) = [0.8808, 0.8808, 0.8808]$$

Then to compute the output values, the values of the previous layer h_{in} should be multiplied by its weights w_{ho} and summed with the bias b_{ho}

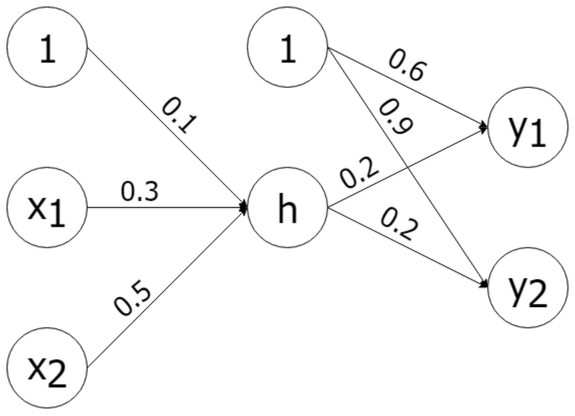
product matrix

$$y_{in} = hw_{ho} + b_{ho} = [1.3212, 1.3212, 1.3212] + [1, 1, 1] = [2.3212, 2.3212, 2.3212]$$

$$y = \sigma(y_{in}) = [0.9106, 0.9106, 0.9106]$$

3 Question

Consider the neural network shown below. It consists of 2 input nodes, 1 hidden node, and 2 output nodes, with addition to a bias at the input layer and a bias at the hidden layer. All nodes in the hidden and output layers use the sigmoid activation function (σ).



(a) Calculate the output of y1 and y2 if the network is fed $x = (1, 0)$ as input.

Solution

$$h_{in} = b_h + w_{x_1\text{-}h}x_1 + w_{x_2\text{-}h}x_2 = (0.1) + (0.3 \times 1) + (0.5 \times 0) = 0.4$$

$$h = \sigma(h_{in}) = \sigma(0.4) = \frac{1}{1 + e^{-0.4}} = 0.599$$

$$y_{1,in} = b_{y_1} + w_{h\text{-}y_1}h = 0.6 + (0.2 \times 0.599) = 0.72$$

$$y_1 = \sigma(0.72) = \frac{1}{1 + e^{-0.72}} = 0.673$$

$$y_{2,in} = b_{y_2} + w_{h\text{-}y_2}h = 0.9 + (0.2 \times 0.599) = 1.02$$

$$y_2 = \sigma(1.22) = \frac{1}{1 + e^{-1.02}} = 0.735$$

As a result, the output is calculated as $y = (y1, y2) = (0.673, 0.735)$.

(b) Assume that the expected output for the input $x = (1, 0)$ is supposed to be $t = (0, 1)$, calculate the updated weights after the backpropagation of the error for this sample. Assume that the learning rate $\eta = 0.1$.

Error for hidden layer + output

Solution

$$\delta_{y_1} = y_1(1 - y_1)(y_1 - t_1) = 0.673(1 - 0.673)(0.673 - 0) = 0.148$$

$$\delta_{y_2} = y_2(1 - y_2)(y_2 - t_2) = 0.735(1 - 0.735)(0.735 - 1) = -0.052$$

$$\delta_h = h(1 - h) \sum_{i=1,2} w_{h\text{-}y_i} \delta_{y_i} = 0.599(1 - 0.599)[0.2 \times 0.148 + 0.2 \times (-0.052)] = 0.005$$

$$\Delta w_{x_1\text{-}h} = -\eta \delta_h x_1 = -0.1 \times 0.005 \times 1 = -0.0005$$

$$\Delta w_{x_2\text{-}h} = -\eta \delta_h x_2 = -0.1 \times 0.005 \times 0 = 0$$

$$\Delta b_h = -\eta \delta_h = -0.1 \times 0.005 = -0.0005$$

$$\Delta w_{h\text{-}y_1} = -\eta \delta_{y_1} h = -0.1 \times 0.148 \times 0.599 = -0.0088652$$

$$\Delta b_{y_1} = -\eta \delta_{y_1} = -0.1 \times 0.148 = -0.0148$$

$$\Delta w_{h\text{-}y_2} = -\eta \delta_{y_2} h = -0.1 \times (-0.052) \times 0.599 = 0.0031148$$

$$\Delta b_{y_2} = -\eta \delta_{y_2} = -0.1 \times (-0.052) = 0.0052$$

Update weights

$$w_{x_1\text{-}h,new} = w_{x_1\text{-}h} + \Delta w_{x_1\text{-}h} = 0.3 + (-0.0005) = 0.2995$$

$$w_{x_2\text{-}h,new} = w_{x_2\text{-}h} + \Delta w_{x_2\text{-}h} = 0.5 + 0 = 0.5$$

$$b_{h,new} = b_h + \Delta b_h = 0.1 + (-0.0005) = 0.0995$$

$$w_{h\text{-}y_1,new} = w_{h\text{-}y_1} + \Delta w_{h\text{-}y_1} = 0.2 + (-0.0088652) = 0.1911348$$

$$b_{y_1,new} = b_{y_1} + \Delta b_{y_1} = 0.6 + (-0.0148) = 0.5852$$

$$w_{h\text{-}y_2,new} = w_{h\text{-}y_2} + \Delta w_{h\text{-}y_2} = 0.2 + 0.0031148 = 0.2031148$$

$$b_{y_2,new} = b_{y_2} + \Delta b_{y_2} = 0.9 + 0.0052 = 0.9052$$

Calculate new weights

4 Question

Recalculate your answers to the previous question, using matrix notation.

Solution

Watch out for matrix notation

First, we need to visualize our parameters in matrix format.

$$x = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad t = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$w_{x\text{-}h} = \begin{bmatrix} 0.3 \\ 0.5 \end{bmatrix} \qquad b_h = \begin{bmatrix} 0.1 \end{bmatrix} \qquad w_{h\text{-}y} = \begin{bmatrix} 0.2 & 0.2 \end{bmatrix} \qquad b_y = \begin{bmatrix} 0.6 & 0.9 \end{bmatrix}$$

(a) In the formulas below, $\sigma(X)$ stands for element-wise sigmoid of matrix X.

b: bias, h: hidden node, y: neuron

$$h_{in} = x w_{x\text{-}h} + b_h = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 0.1 \end{bmatrix} = \begin{bmatrix} 0.4 \end{bmatrix}$$

$$h = \sigma(h_{in}) = \sigma\left(\begin{bmatrix} 0.4 \end{bmatrix}\right) = \begin{bmatrix} 0.599 \end{bmatrix}$$

$$y_{in} = h w_{h\text{-}y} + b_y = \begin{bmatrix} 0.599 \end{bmatrix} \begin{bmatrix} 0.2 & 0.2 \end{bmatrix} + \begin{bmatrix} 0.6 & 0.9 \end{bmatrix} = \begin{bmatrix} 0.72 & 1.02 \end{bmatrix}$$

$$y = \sigma\left(\begin{bmatrix} 0.72 & 1.02 \end{bmatrix}\right) = \begin{bmatrix} 0.673 & 0.735 \end{bmatrix}$$

(b) In the formulas below, $X \circ Y$ stands for element-wise multiplication of matrices X and Y, and $1 - X$ is equivalent to subtracting the matrix X from a matrix with the same size of X, but with all elements equal to 1.

$$\delta_y = y \circ (1 - y) \circ (y - t) =$$

$$\begin{bmatrix} 0.673 & 0.735 \end{bmatrix} \circ \left(1 - \begin{bmatrix} 0.673 & 0.735 \end{bmatrix}\right) \circ \left(\begin{bmatrix} 0.673 & 0.735 \end{bmatrix} - \begin{bmatrix} 0 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0.148 & -0.052 \end{bmatrix}$$

$$\boxed{\delta_h} = h \circ (1 - h) \circ (\delta_y w_{h\text{-}y}^T) =$$

$$\begin{bmatrix} 0.599 \end{bmatrix} \circ \left(1 - \begin{bmatrix} 0.599 \end{bmatrix}\right) \circ \left(\begin{bmatrix} 0.148 & -0.052 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}\right) = \boxed{\begin{bmatrix} 0.005 \end{bmatrix}}$$

$$\Delta w_{x\text{-}h} = -\eta(x^T \boxed{\delta_h}) = -0.1 \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0.005 \end{bmatrix}\right) = \begin{bmatrix} -0.0005 \\ 0 \end{bmatrix}$$

$$\Delta b_h = \boxed{-\eta \delta_h} = -0.1 \begin{bmatrix} 0.005 \end{bmatrix} = \begin{bmatrix} -0.0005 \end{bmatrix}$$

$$\Delta w_{h\text{-}y} = -\eta(h^T \delta_y) = -0.1 \left(\begin{bmatrix} 0.599 \end{bmatrix} \begin{bmatrix} 0.148 & -0.052 \end{bmatrix}\right) = \begin{bmatrix} -0.00886 & 0.00311 \end{bmatrix}$$

$$\Delta b_y = -\eta \delta_y = -0.1 \begin{bmatrix} 0.148 & -0.052 \end{bmatrix} = \begin{bmatrix} -0.0148 & 0.0052 \end{bmatrix}$$

$$w_{x\text{-}h,new} = w_{x\text{-}h} + \Delta w_{x\text{-}h} = \begin{bmatrix} 0.3 \\ 0.5 \end{bmatrix} + \begin{bmatrix} -0.0005 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2995 \\ 0.5 \end{bmatrix}$$

$$b_{h,new} = b_h + \Delta b_h = \begin{bmatrix} 0.1 \end{bmatrix} + \begin{bmatrix} -0.0005 \end{bmatrix} = \begin{bmatrix} 0.0995 \end{bmatrix}$$

$$w_{h\text{-}y,new} = w_{h\text{-}y} + \Delta w_{h\text{-}y} = \begin{bmatrix} 0.2 & 0.2 \end{bmatrix} + \begin{bmatrix} -0.0088652 & 0.0031148 \end{bmatrix} = \begin{bmatrix} 0.1911 & 0.2031 \end{bmatrix}$$

$$b_{y,new} = b_y + \Delta b_y = \begin{bmatrix} 0.6 & 0.9 \end{bmatrix} + \begin{bmatrix} -0.0148 & 0.0052 \end{bmatrix} = \begin{bmatrix} 0.5852 & 0.9052 \end{bmatrix}$$

COMP 472: Artificial Intelligence
State Space Representation and Uninformed Search
Solutions

1 Question

Consider the following problem:

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.

Solution

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.

Solution

`position(f,f,f,f)`

(c) Write down all illegal states.

Solution

```
 position(f,o,o,o)
position(f,o,o,f)
position(o,f,f,o)
position(o,f,f,f)
position(f,f,o,o)
position(o,o,f,f)
position(o,f,f,f)
```

(d) Write down the possible actions.

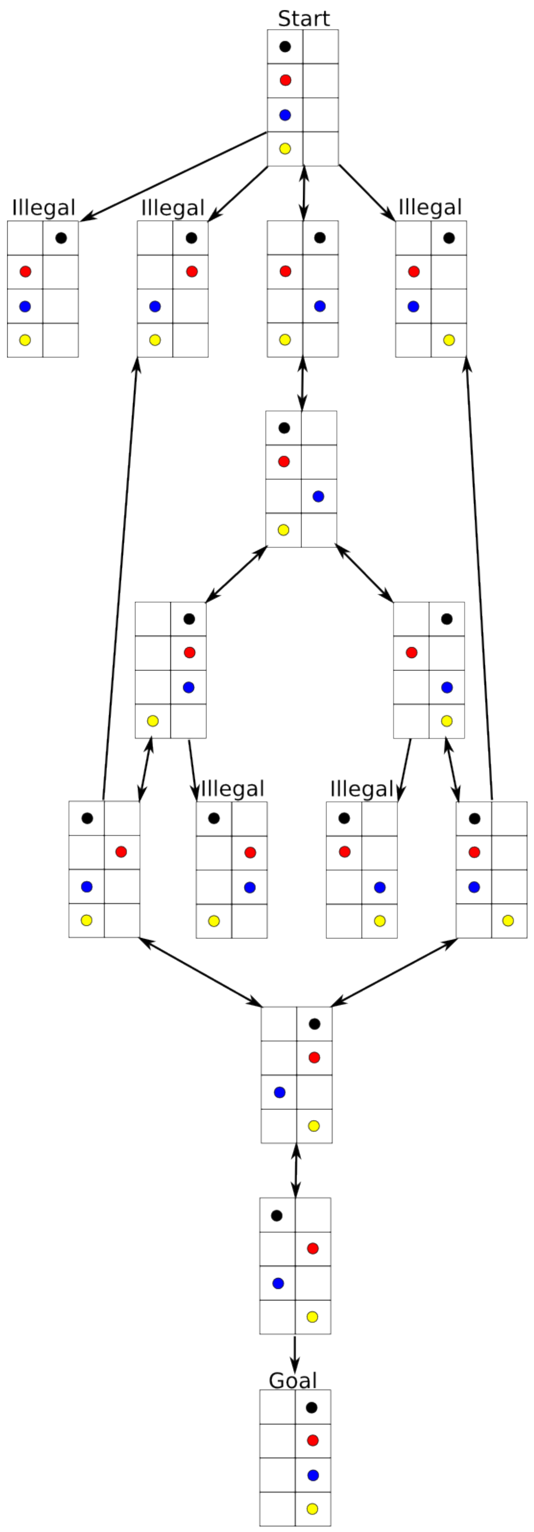
Solution

```
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.

Solution

To make it easier, the figure below represents each state graphically. The first column is the original bank `o`, the second column is the far back `f`; and the colored dots represent an entity (black → farmer, red → fox, blue → goose, and yellow → beans). However, we could have used the predicate `position` instead. For example, the initial state is equivalent to `position(o,o,o,o)` and the goal state is equivalent to `position(f,f,f,f)`.


Start
Illegal
Illegal
Illegal
Illegal
Illegal
Goal

(f) Find a sequence of moves to solve this problem.

Solution

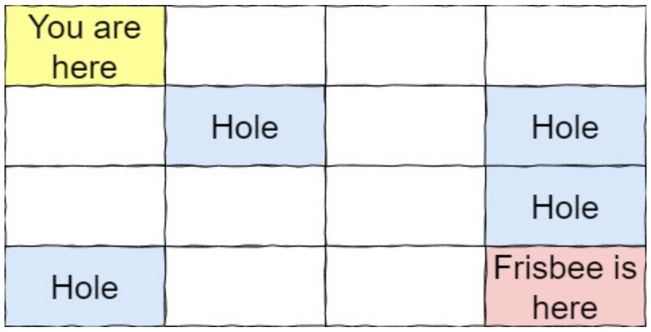
```
moveFar(farmer, goose),
moveBack(farmer, empty),
moveFar(farmer, fox),
moveBack(farmer, goose),
moveFar(farmer, beans),
moveBack(farmer, empty),
moveBack(farmer, goose)
```

OR

```
moveFar(farmer, goose),
moveBack(farmer, empty),
moveFar(farmer, beans),
moveBack(farmer, goose),
moveFar(farmer, fox),
moveBack(farmer, empty),
moveBack(farmer, goose)
```

2 Question

Exercise from OpenAI[1]*:* 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:



(a) Let a 4×4 matrix represent the above grid. The position of each cell in the grid can then be represented by the indices of the elements of the matrix. Given this representation, write down the initial state and the goal states.

Solution

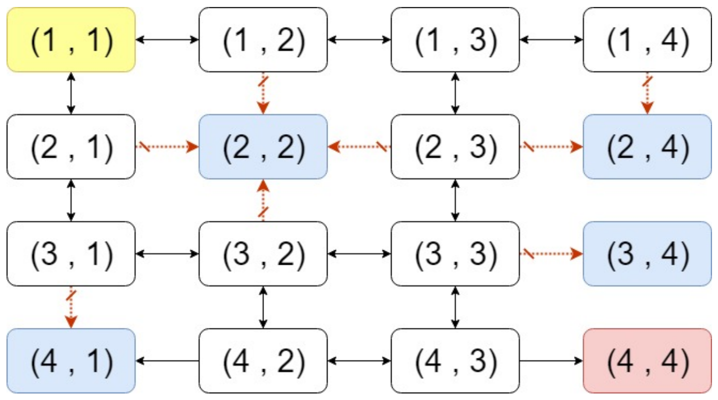
- Initial state: $(1, 1)$
- Goal state: $(4, 4)$

(b) Assume that 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.
Draw the state space for this problem.

Solution

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.

[1] `https://gym.openai.com/envs/FrozenLake-v0/`



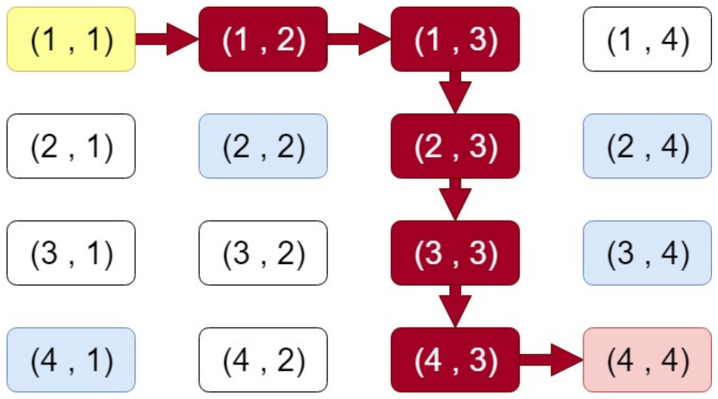
(c) Find a sequence of moves to solve this problem.

Solution

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

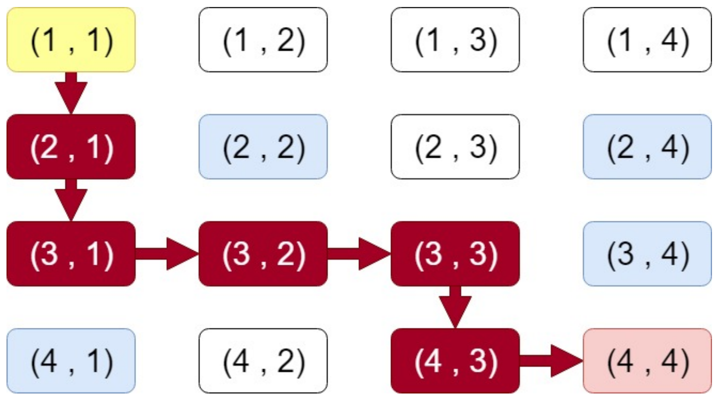
Solution Path 1

Solution Path 1: $(1,1) \to (1,2) \to (1,3) \to (2,3) \to (3,3) \to (4,3) \to (4,4)$
Cost of solution path: 6 (since each move has a cost of 1)



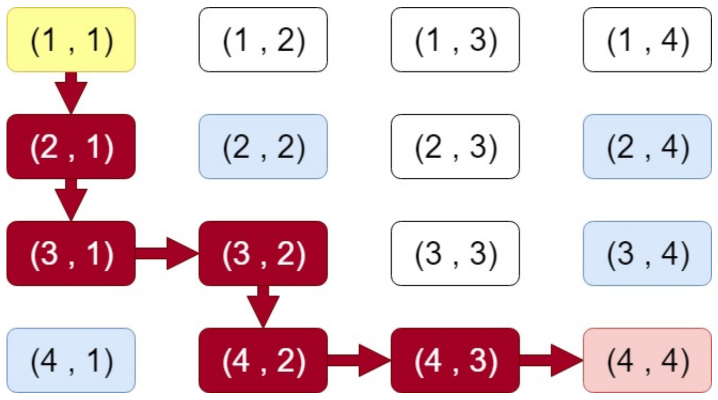
Solution Path 2

Solution Path 2: $(1,1) \to (2,1) \to (3,1) \to (3,2) \to (3,3) \to (4,3) \to (4,4)$
Cost of solution path: 6 (since each move has a cost of 1)



Solution Path 3

Solution Path 3: $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (4,2) \rightarrow (4,3) \rightarrow (4,4)$
Cost of solution path: 6 (since each move has a cost of 1)



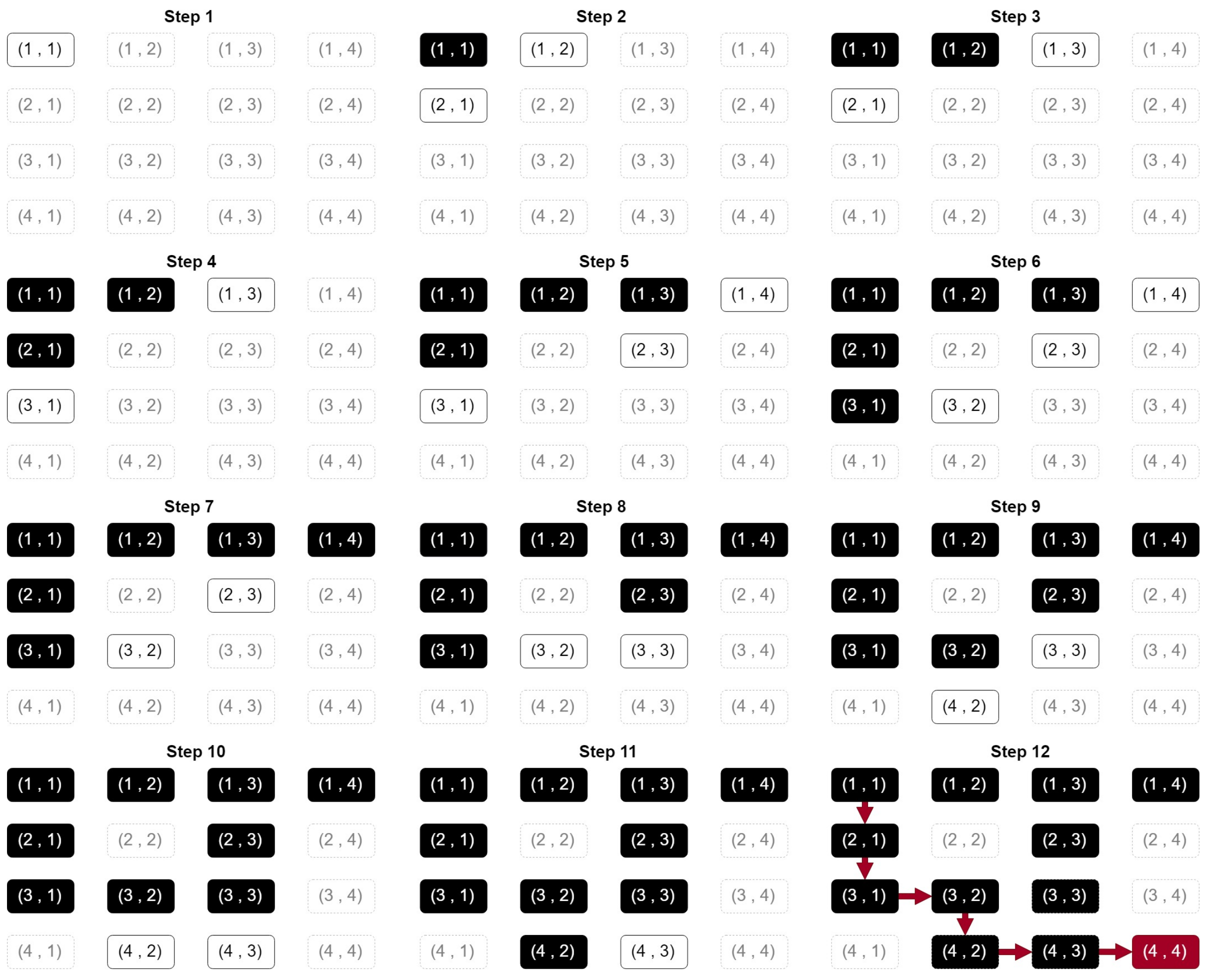
(d) Perform a Breadth-First Search on this state space. At each step, indicate the content of the OPEN and CLOSED lists.

Solution

The trace of Breadth-First Search is listed below.

Step	Visited	OPEN	CLOSED
1		$(1,1)$	$\emptyset$
2	$(1,1)$	$(1,2),(2,1)$	$(1,1)$
3	$(1,2)$	$(2,1),(1,3)$	$(1,2),(1,1)$
4	$(2,1)$	$(1,3),(3,1)$	$(2,1),(1,2),(1,1)$
5	$(1,3)$	$(3,1),(1,4),(2,3)$	$(1,3),(2,1),(1,2),(1,1)$
6	$(3,1)$	$(1,4),(2,3),(3,2)$	$(3,1),(1,3),(2,1),(1,2),(1,1)$
7	$(1,4)$	$(2,3),(3,2)$	$(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
8	$(2,3)$	$(3,2),(3,3)$	$(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
9	$(3,2)$	$(3,3),(4,2)$	$(3,2),(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
10	$(3,3)$	$(4,2),(4,3)$	$(3,3),(3,2),(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
11	$(4,2)$	$(4,3)$	$(4,2),(3,3),(3,2),(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
12	$(4,3)$	$(4,4)$	$(4,3),(4,2),(3,3),(3,2),(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$
13	$(4,4)$		$(4,4),(4,3),(4,2),(3,3),(3,2),(2,3),(1,4),(3,1),(1,3),(2,1),(1,2),(1,1)$

The goal state (4,4) is visited, so the search terminates.



Progress of Breadth-First Search. 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.

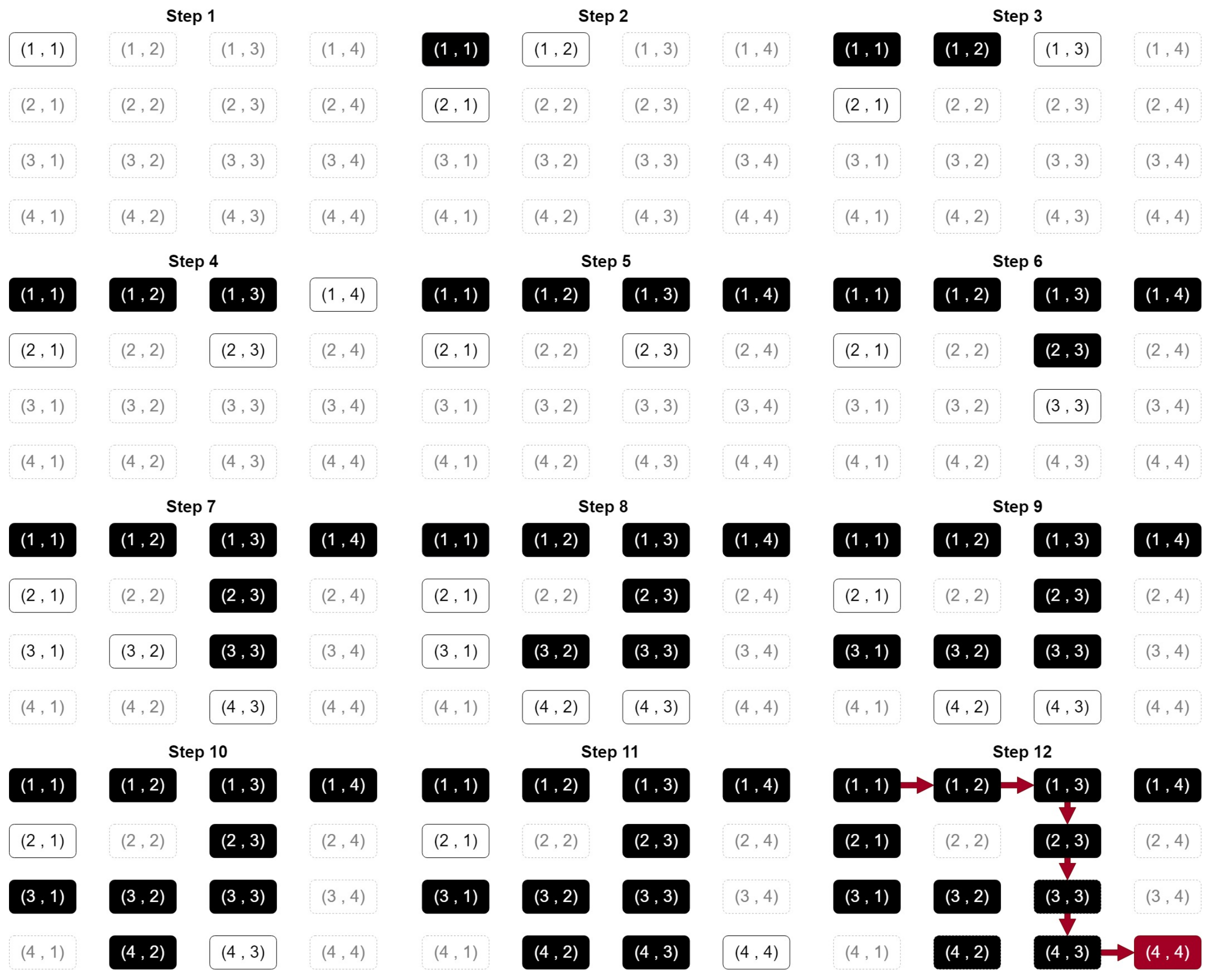
(e) Perform a Depth-First Search on this state space. At each step, indicate the content of the OPEN and CLOSED lists.

Solution

The trace of Depth-First Search is listed below.

Step	Visited	OPEN	CLOSED
1		(1,1)	∅
2	(1, 1)	(1, 2), (2, 1)	(1, 1)
3	(1, 2)	(1, 3), (2, 1)	(1, 2), (1, 1)
4	(1, 3)	(1, 4), (2, 3), (2, 1)	(1, 3), (1, 2), (1, 1)
5	(1, 4)	(2, 3), (2, 1)	(1, 4), (1, 3), (1, 2), (1, 1)
6	(2, 3)	(3, 3), (2, 1)	(2,3), (1,4), (1,3), (1,2), (1,1)
7	(3, 3)	(3, 2), (4, 3), (2, 1)	(3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)
8	(3, 2)	(3, 1), (4, 2), (4, 3), (2, 1)	(3,2), (3,3), (2,3), (1,4), (1,3), (1,2), (1,1)
9	(3, 1)	(4, 2), (4, 3), (2, 1)	(3,1), (3,2), (3,3), (2,3), (1,4), (1,3), (1,2), (1,1)
10	(4, 2)	(4, 3), (2, 1)	(4, 2), (3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)
11	(4, 3)	(4, 4), (2, 1)	(4, 3), (4, 2), (3, 1), (3, 2), (3, 3), (2, 3), (1, 4), (1, 3), (1, 2), (1, 1)
12	(4, 4)	(2, 1)	(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 (4,4) is visited, so the search terminates.



Progress of Depth-First Search. 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.

(f) Perform a Uniform Cost Search on this state space. At each step, indicate the content of the OPEN and CLOSED lists.

Solution

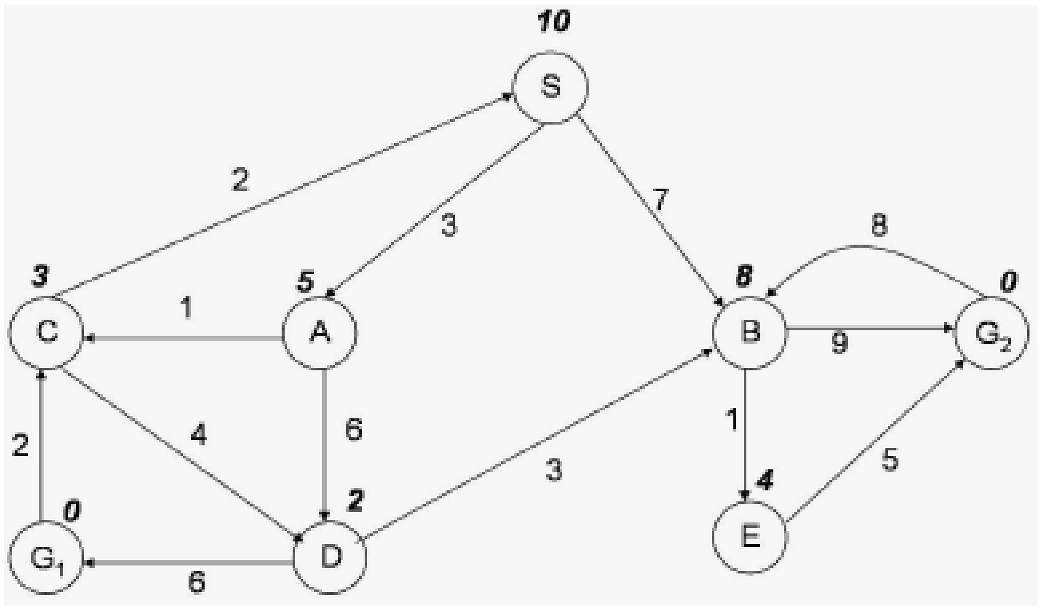
Since we have assumed that all move costs are equal, then Uniform-Cost Search and Breadth-First Search will have the same result.

COMP 472: Artificial Intelligence

State Space Search

Solutions

Question 1 Consider the state space below. 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: The depth of each state in the open list is indicated 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) General 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) $\not<$ D(2); G1(0) $<$ D(2)
5	G1	

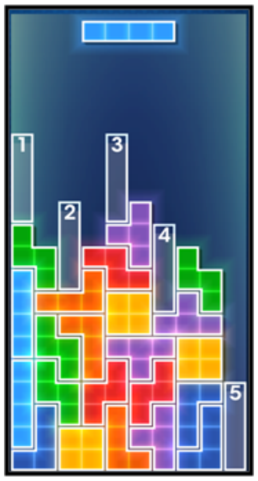
(f) Best-first search

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 2 Consider the classical game of Tetris. The objective of the game is to move and rotate each falling block so as to create an entire row of block pieces without any gap. If such a row is created, then that row disappears, and all rows on top of it fall down.



For example, in the figure above, a 1x4 block is falling from the top. The player can move this block left and right and rotate it by 90 degree to have it upright. As the figure shows, the player has at least 5 choices to position the block. Out of these 5 choices, position 5 is preferable, because then the 4 bottom rows would be complete. These rows would then disappear; all rows above would fall down, leaving more space on top to place the next random block to fall. The game ends when too many blocks are stacked up (no new complete rows can be created) and no more blocks can be placed on the board.

(a) Formulate a simple heuristic to determine how to place a falling (random) block on an existing board.

A possible heuristic is to count the number of complete rows created (that will be deleted). In that case, h(option2) = 0 and h(option4) = 0.

Another heuristic is to count the number of "touching sides" i.e. how many sides of the falling block will be in contact with the Tetris pile. In that case, h(option2) = 5 (2 edges on the left + the bottom + 2 edges on the right) h(option4) = 8 (4 edges on the left + the bottom + 3 edges on the right)

There are plenty of other possible heuristics.

(b) Now apply your heuristic to evaluate option 2 and option 4 of the figure above.

COMP 472: Artificial Intelligence

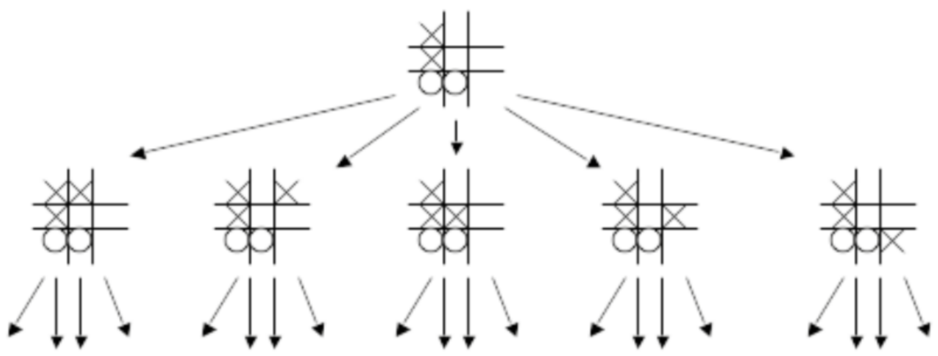
Minimax and Alpha-Beta Pruning

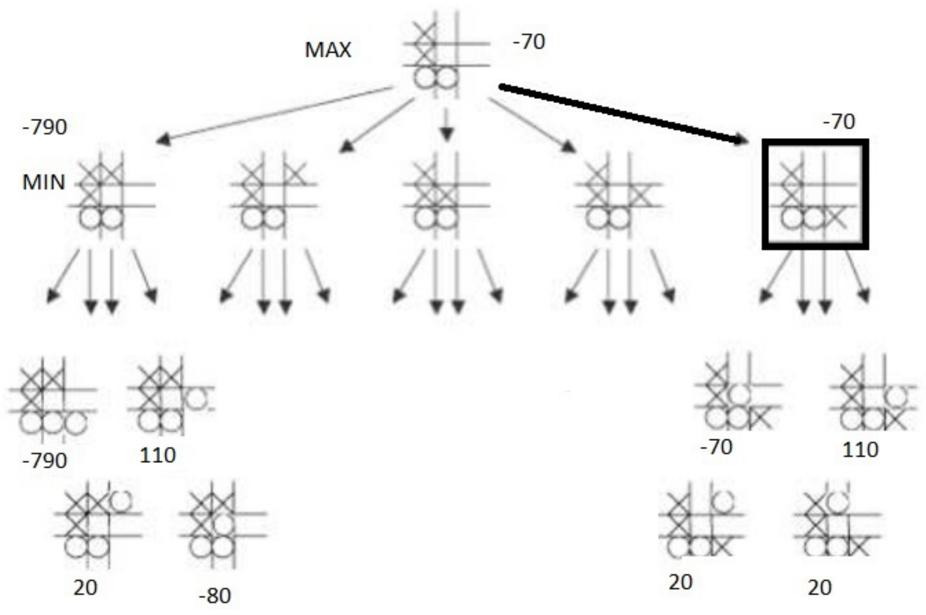
Solutions

Question 1 Consider state space search for the game of Tic-Tac-Toe. You are the X player, looking at the board shown below, with five possible moves. You want to look ahead to find your best move and decide to use the following evaluation function for rating board configurations:

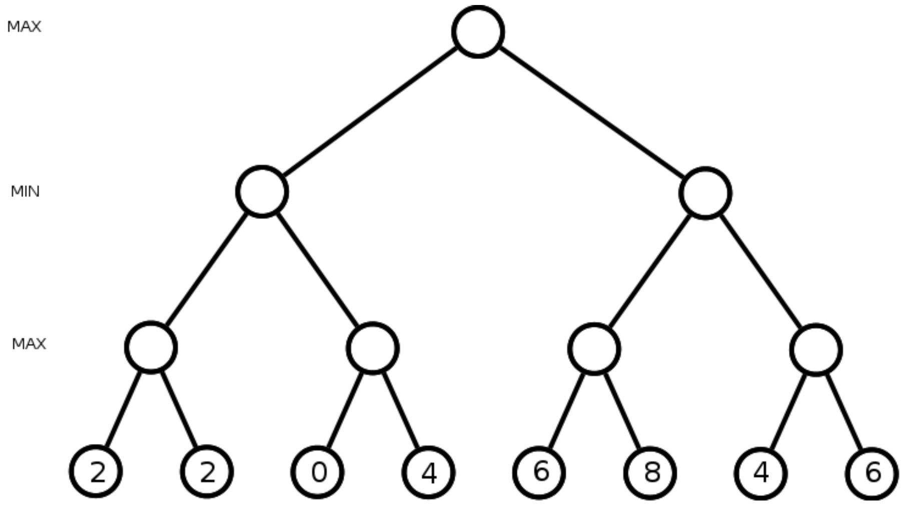
```
value $V = 0$
for all rows, columns, diagonals $R$ do:
    if $R$ contains three $X$s then:
        $V = V + 1000$
    else if $R$ contains three 0s then:
        $V = V - 1000$
    else if $R$ contains two $X$s then:
        $V = V + 100$
    else if $R$ contains two $O$s then:
        $V = V - 100$
    else if $R$ contains one $X$ then:
        $V = V + 10$
    else if $R$ contains one $O$ then:
        $V = V - 10$
    end if
end for
return $V$
```

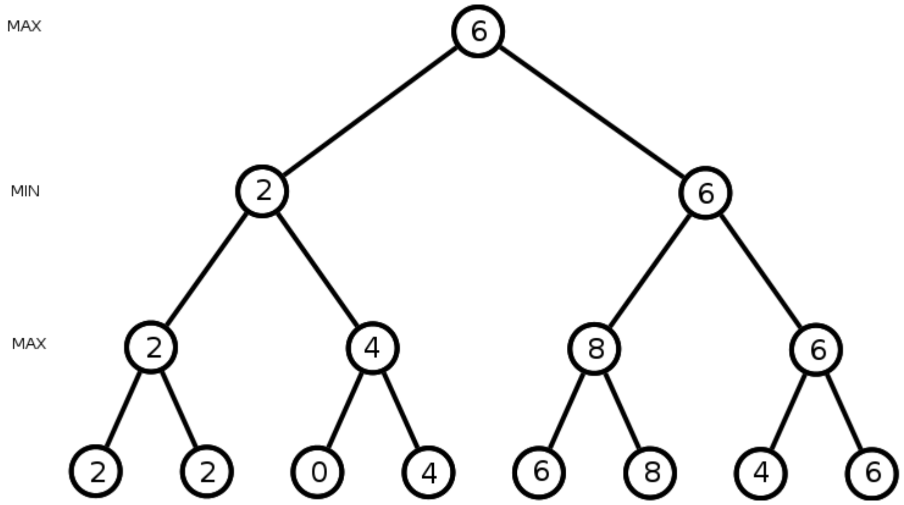
Draw the four possible configurations for the leftmost and the rightmost board configurations below. Use the evaluation function above to rate these 8 board configurations and choose X's best move.



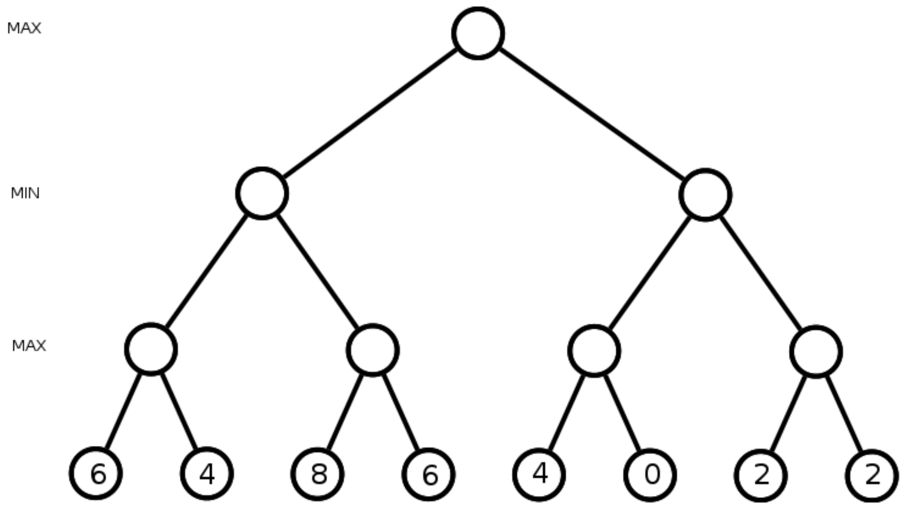

MAX
-70
-790
MIN
-70
-790
110
20
-80
-70
110
20
20

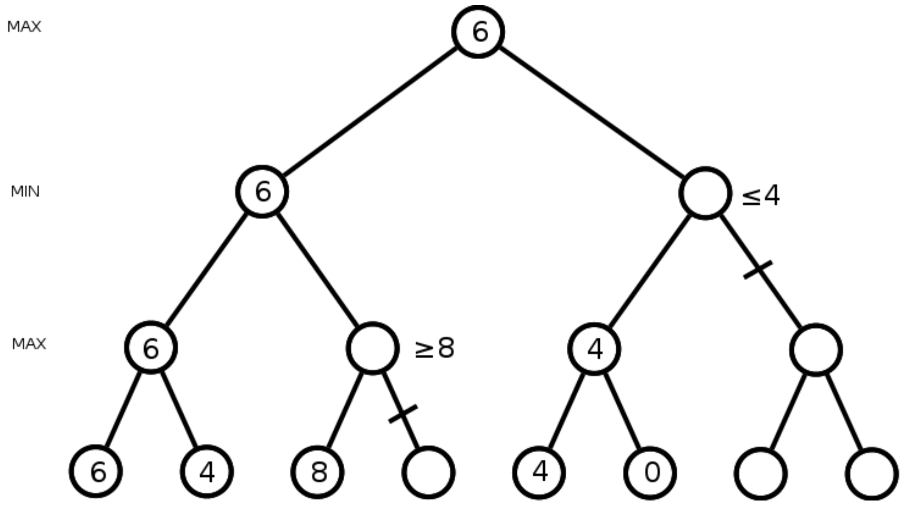
Question 2 (a) Consider the game tree shown below. Explore the tree using Alpha-Beta. Indicate all parts of the tree that are pruned, and indicate the winning path or paths.





(b) Now do the same for the tree below, which is a mirror image of the tree shown above.

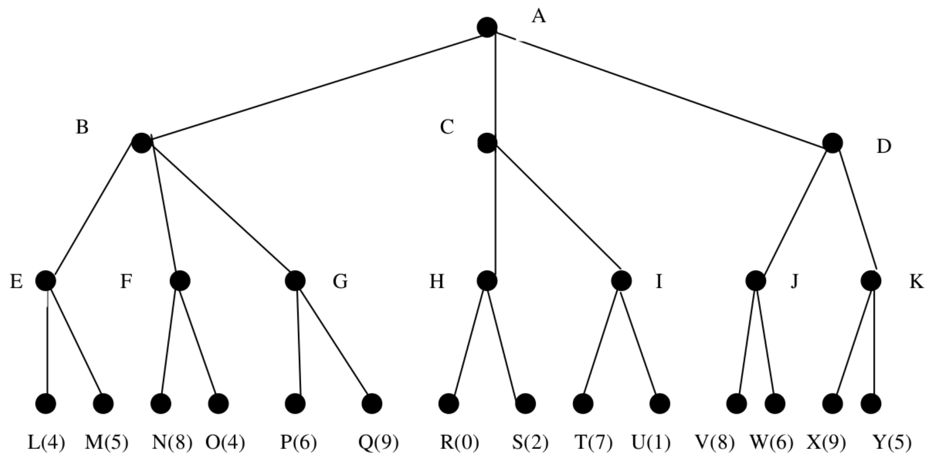




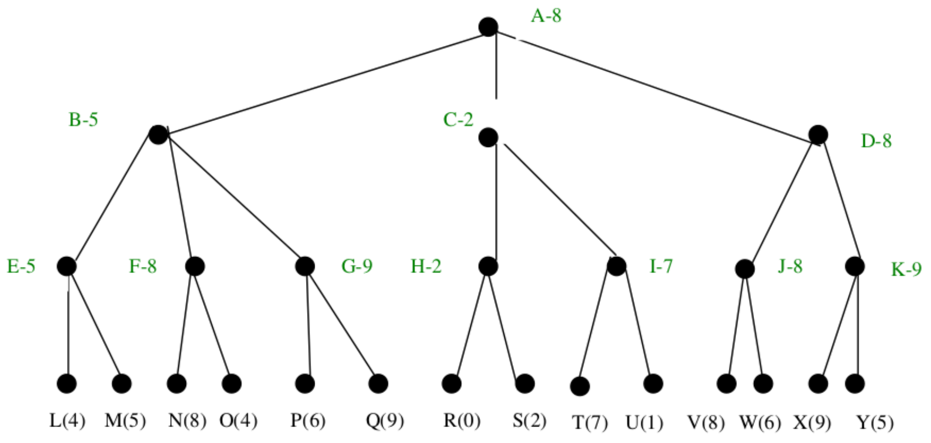
(c) Compare the amount of pruning in the above two trees. What do you notice about how the order of evaluation nodes affects the amount of Alpha-Beta pruning? *If the evaluated nodes are ordered in the manner described below, then alpha-beta gets maximal pruning.*

You get maximal cutoff if the left-most descendent of a MAX node has the largest $e(n)$ *value compared to its siblings. For a MIN level, you get maximal pruning if the left-most descendent has the lowest* $e(n)$ *value compared to its siblings.*

Question 3 Consider the game tree below. Each node is labelled with a letter, and the evaluation function for each leaf is indicated in parentheses. Assume that the MAX player goes first.

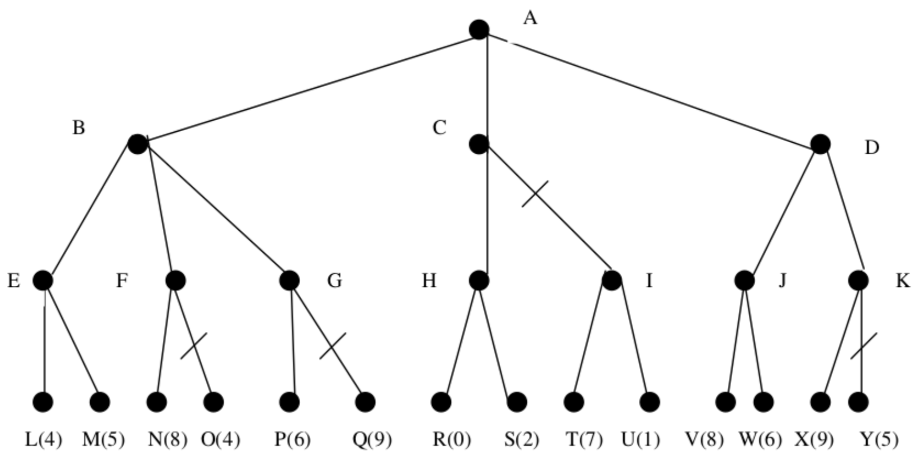


(a) Compute the minimax game value of nodes A, B, C, and D using the minimax algorithm. Show all values that are brought up to the internal nodes. What should MAX do?

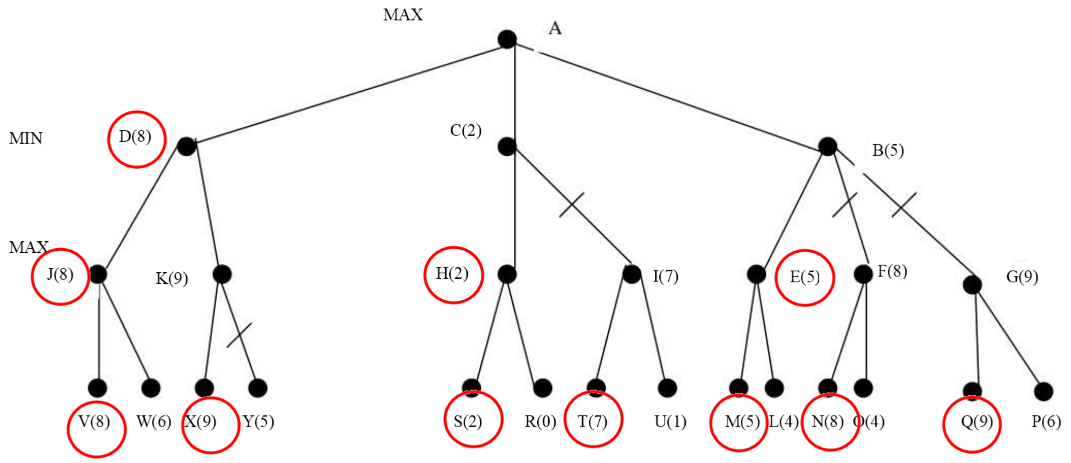


MAX should go right towards state D.

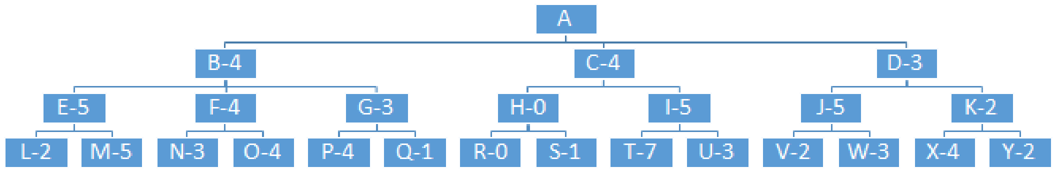
(b) Cross out the branches of all the nodes that are *not* visited by alpha-beta pruning. Show all your work.



(c) Draw a new game tree by re-ordering the nodes, such that the new game tree is equivalent to the tree above, but alpha-beta pruning will prune as many nodes as possible. *This is an optimal tree, other trees could also be optimal for pruning.*



Question 4 Consider the following game tree.



The value of the evaluation function at each node is shown next to its name. For example, B-4 indicates that node B has an evaluation function of 4. All evaluations are from the point of view of the first player.

(a) Assume that the first player is the maximizing player MAX and she looks that all levels (ie, to the level labeled L, M, N, O, . . .). List in order the states that will **NOT** be examined when using alpha-beta pruning.

(b) What move should MAX choose?

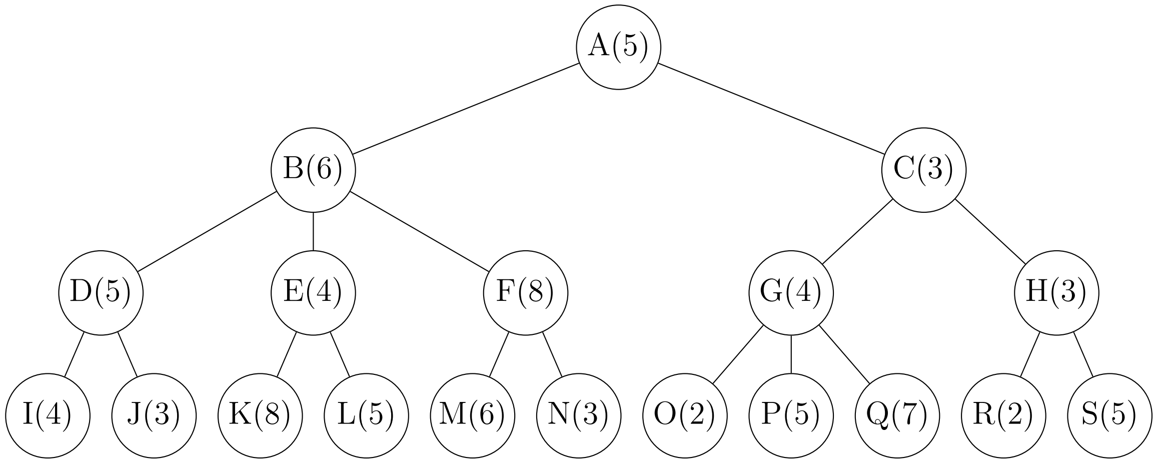
(c) Suppose that instead of looking down all levels, MAX can only afford to look at level 2 (ie, the level with E, F, G, H, . . . instead of the level with L, M, N, O. . .). In theory, could that change MAX's move? Explain.

(a) *Nodes that will not be visited are: Q I T U K Y X*

(b) *MAX should choose to move left (node B)*

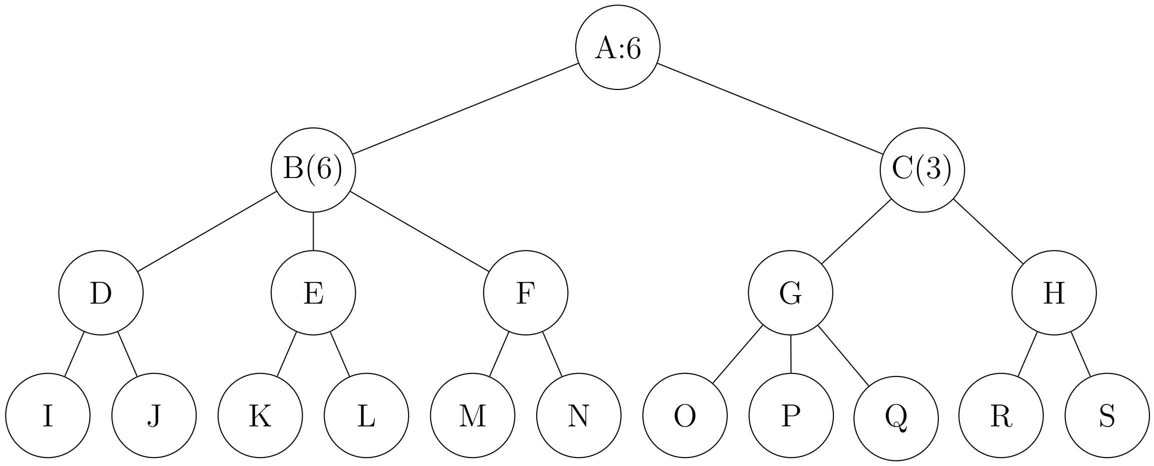
(c) *Yes. MAX's best move could change. The deeper a player can afford to look ahead, the more informed will be his decision.*

Question 5 Consider tree below, in each node the value of the heuristic function is indicated inside parentheses. Assume Max plays first.



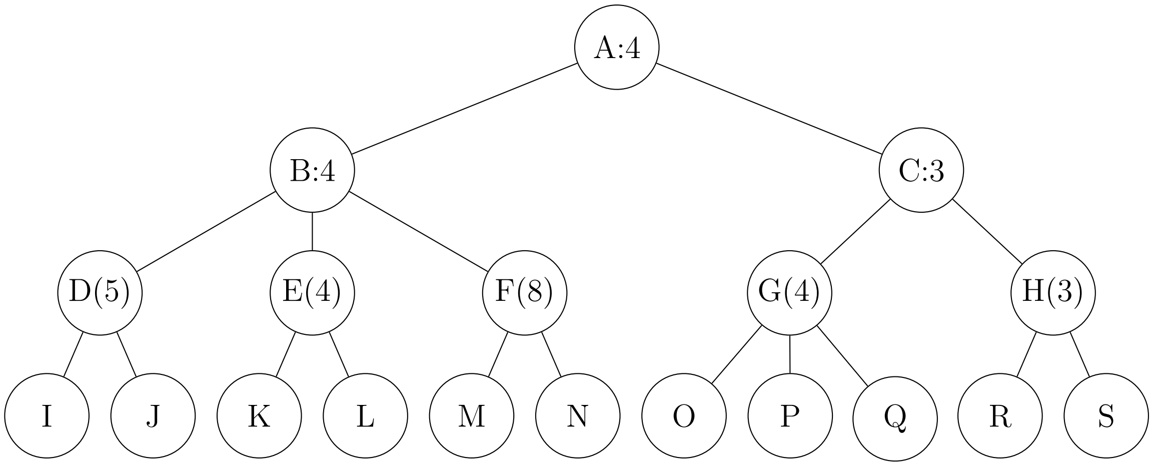
What should Max do first if:

(a) if Max can explore only one level.



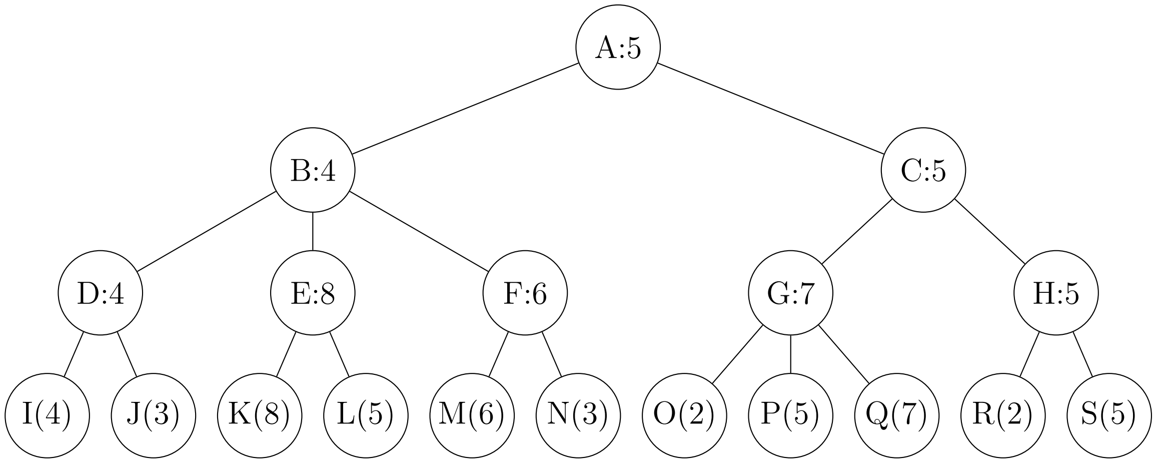
Max should go to B

(b) if Max can explore two levels.



Max should go to B

(c) if Max can explore three levels.



Max should go to C