Problem One: Evaluation Functions

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the evaluation function described in this lecture on Slide 14.
Game Tree Search Practice Problems

Problem One: Evaluation Functions
(SOLUTION)

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the evaluation function described in this lecture on Slide 14.

<table>
<thead>
<tr>
<th>X</th>
<th>O</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: 2</td>
<td>O: -5</td>
<td>Sum: -3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: 3</td>
<td>O: -3</td>
<td>Sum: 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>X</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: 5</td>
<td>O: -2</td>
<td>Sum: 3</td>
</tr>
</tbody>
</table>
Game Tree Search Practice Problems

Problem Two: Evaluation Functions

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the evaluation function described in this lecture on Slide 14.

1. 

2. 

3. 

4.
Game Tree Search Practice Problems

Problem Two: Evaluation Functions

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the evaluation function described in this lecture on Slide 14.

NOTE: We just flipped X’s and O’s, so you should have the negatives of the sums you got in problem one!

<table>
<thead>
<tr>
<th>Board</th>
<th>X: 5</th>
<th>O: -2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sum: 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X: 3</td>
<td>O: -3</td>
</tr>
<tr>
<td></td>
<td>Sum: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X: 2</td>
<td>O: -5</td>
</tr>
<tr>
<td></td>
<td>Sum: -3</td>
<td></td>
</tr>
</tbody>
</table>
Problem Three: Evaluation Functions

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the following evaluation function:

For each sequence of 3 pieces (X, O, or blank) in a row, column, or diagonal:

- Count 1 for 1 X and 2 blanks, and -1 for 1 O and 2 blanks;
- Count 4 for 2 X’s and 1 blank, and -4 for 2 O’s and 1 blank;
- Count 0 for anything else (all blanks, or at least one X and at least one O);
- Return the sum of all such counts.

Board 1:

```
X
O O
```

Board 2:

```
X X O
X
O
```

Board 3:

```
X X
X O
```

Board 4:

```
X X
X O
```

Board 5:

```
X
O O
```

Board 6:

```
X X O
X
O
```

Board 7:

```
X X
X O
```

Board 8:

```
X
O O
```
Game Tree Search Practice Problems

Problem Three: Evaluation Functions
(SOLUTION)

For each of the Tic-Tac-Toe boards shown at right, give the value that would be returned by the following evaluation function:

For each sequence of 3 pieces (X, O, or blank) in a row, column, or diagonal:

- Count 1 for 1 X and 2 blanks, and -1 for 1 O and 2 blanks;
- Count 4 for 2 X’s and 1 blank, and -4 for 2 O’s and 1 blank;
- Count 0 for anything else (all blanks, or at least one X and at least one O);
- Return the sum of all such counts.

![Board 1] X: 2
O: -7
Sum: -5

![Board 2] X: 3
O: -5
Sum: -2

![Board 3] X: 9
O: -2
Sum: 7
**Problem Four:** Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problems on Game Tree Search

**Problem Four**

**Solution:** Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problem Five: Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problem Five

Solution: Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problems on Game Tree Search

**Problem Six:** Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.

![Game Tree Diagram]

Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problem Six
SOLUTION: Given the values returned by the evaluation function for the leaves shown in the diagram, fill in the rest of the values at each node, and state which move would be taken by the root.
Problem Seven

Alpha-Beta Problems. This MinMax tree is the same as problem Four in the previous lecture; the values shown at right are those calculated in a Postorder traversal of the tree.

We will go through the tree Postorder and consider what alpha and beta values are calculated, and when cut-offs occur.
Practice problems on Alpha-Beta Pruning

Problem Seven

*Alpha* values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

*Beta* values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

**Problem One:** The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?
Practice problems on Alpha-Beta Pruning

Alpha values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

Beta values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

Problem One: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the beta value “≤ B” at the red node before continuing the Postorder traversal into the right child?

Solution as shown.
Practice problems on Alpha-Beta Pruning

**Alpha** values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

**Beta** values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

**Problem One:** The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the red node before continuing the Postorder traversal into the right child?
Practice problems on Alpha-Beta Pruning

**Alpha** values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

**Beta** values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

**Problem One:** The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the red node before continuing the Postorder traversal into the right child?

Solution as shown.
Practice problems on Alpha-Beta Pruning

Alpha values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

Beta values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

Problem One: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the red node before continuing the Postorder traversal into the right child?

Solution as shown.

(C) Will a cut-off occur at this point so that the right child of the red node is not visited?
Practice problems on Alpha-Beta Pruning

**Alpha** values are shown at max nodes in the form “≥ A” meaning “at this point in the traversal, I know I can get A or more”

**Beta** values are shown at min nodes in the form “≤ B” meaning “at this point in the traversal, I know I can get B or less.”

**Problem One:** The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the red node before continuing the Postorder traversal into the right child?

Solution as shown.

(C) Will a cut-off occur at this point so that the right child of the red node is not visited?

Solution: NO! It is possible that a value at that node is ≤ 5 and ≥ -1. In fact, this happens (see next slide).
Problem 8: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the node A before continuing the Postorder traversal into the right child?
Problem Eight: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

Solution as shown.
**Problem Eight:** The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “$\leq B$” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “$\geq A$” at the red node before continuing the Postorder traversal into the right child? [Hint: this would have been inherited from the root.]
Problem Eight: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value “≤ B” at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the red node before continuing the Postorder traversal into the right child? [Hint: this would have been inherited from the root.]

Solution: ≥ 5
**Problems on Alpha-Beta Pruning**

**Problem Eight:** The tree at right shows the calculation after returning from the leftmost child of the node A.

(A) What is the alpha value “≤ B” at the node A before continuing the Postorder traversal into the right child?

(B) What is the alpha value “≥ A” at the node A before continuing the Postorder traversal into the right child? [Hint: this would have been inherited from the root.]

(C) Would a cut-off occur before going into the right child of the node A?
Problem Eight: The tree at right shows the calculation after returning from the leftmost child of the red node.

(A) What is the alpha value \( \leq B \) at the red node before continuing the Postorder traversal into the right child?

(B) What is the alpha value \( \geq A \) at the red node before continuing the Postorder traversal into the right child? [Hint: this would have been inherited from the root.]

(C) Would a cut-off occur before going into the right child of the red node?

Solution: YES! There is no possible way for a value to be both \( \leq -2 \) and \( \geq 5 \).