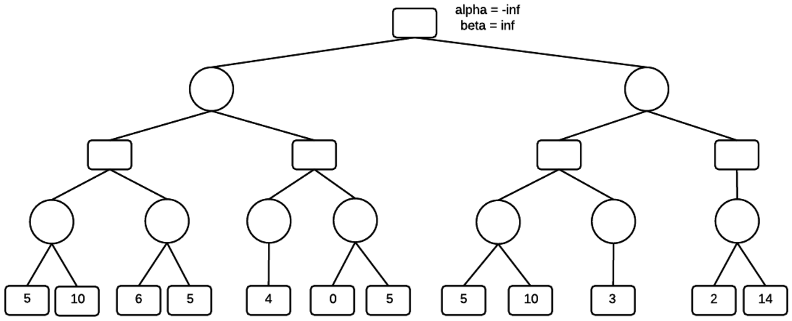
CSCI 3202 - Introduction to Artificial Intelligence
Instructor: Hoenigman
Assignment 4
Due Friday, September 23 by 4pm.
Submit written answers at the beginning of class

Problems

1. Use the alpha-beta algorithm to show the minimax value to MAX in the following game tree. Show the branches that are pruned during the search, and the values for MAX and MIN at each level in the tree.



Rubric: 40 points total

Feature	**Points**
Branches pruned - 10 points for each branch. There should be two branches pruned.	20
Correct values assigned for each MIN/MAX node - 1 point for each value	14
Alpha, beta values are passed down from the parent node. Beta updated at MIN node only. Alpha updated at MAX node only. Subtract 1/2 point for incorrect values, maximum deduction of 6 points.	6

2. Show that the following assertion is true using an example: For every game tree, the utility obtained by MAX using minimax decisions against a suboptimal MIN will never be lower than the utility obtained playing against an optimal MIN. Your example should also include an explanation of what an

optimal utility is for MAX and MIN and what it would mean for either MAX or MIN to play a sub-optimal game.

Answer: A suboptimal MIN means that MIN doesn't select the minimum value from its children and a suboptimal MAX means that MAX doesn't select the maximum value from its children. If MIN selects suboptimal value, then the choice for MAX, as the parent of MIN, will be at least as big as if MIN were playing optimally. The same is true for MIN as the parent of MAX, where if MAX selects a suboptimal value, then MIN will have a choice that is less than or equal to an optimal value.

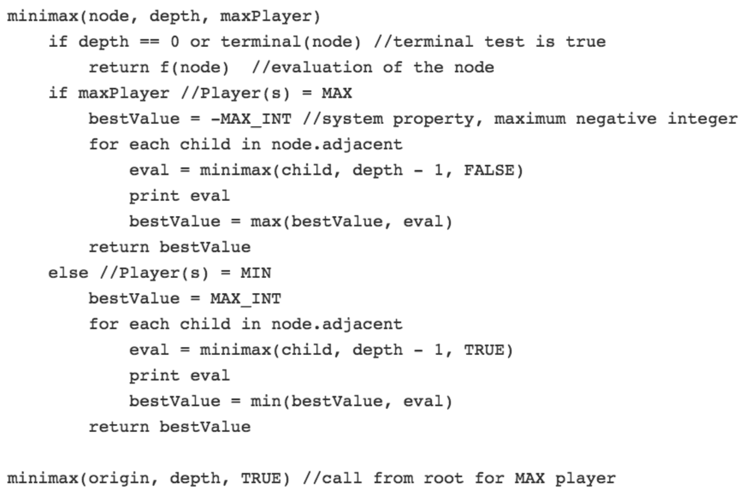
Rubric: 30 points total

Feature	Rubric
Definition of suboptimal MIN and MAX.	15
Example, either written or drawn showing either a suboptimal MIN play or a suboptimal MAX play.	15

3. The following pseudo-code for the minimax algorithm is similar to the code included in the lecture notes. This code shows additional print commands that are not in the lecture notes code.

Rubric: 30 points total

Feature	Points
Leaf nodes printed left to right.	10
Parent nodes printed after child nodes.	10
Root not printed.	10



```
minimax(node, depth, maxPlayer)
    if depth == 0 or terminal(node) //terminal test is true
        return f(node)  //evaluation of the node
    if maxPlayer //Player(s) = MAX
        bestValue = -MAX_INT //system property, maximum negative integer
        for each child in node.adjacent
            eval = minimax(child, depth - 1, FALSE)
            print eval
            bestValue = max(bestValue, eval)
        return bestValue
    else //Player(s) = MIN
        bestValue = MAX_INT
        for each child in node.adjacent
            eval = minimax(child, depth - 1, TRUE)
            print eval
            bestValue = min(bestValue, eval)
        return bestValue

minimax(origin, depth, TRUE) //call from root for MAX player
```

For the following tree, what numbers are displayed when the algorithm executes with an initial depth = 4. You can assume that the root of the tree only has one child.