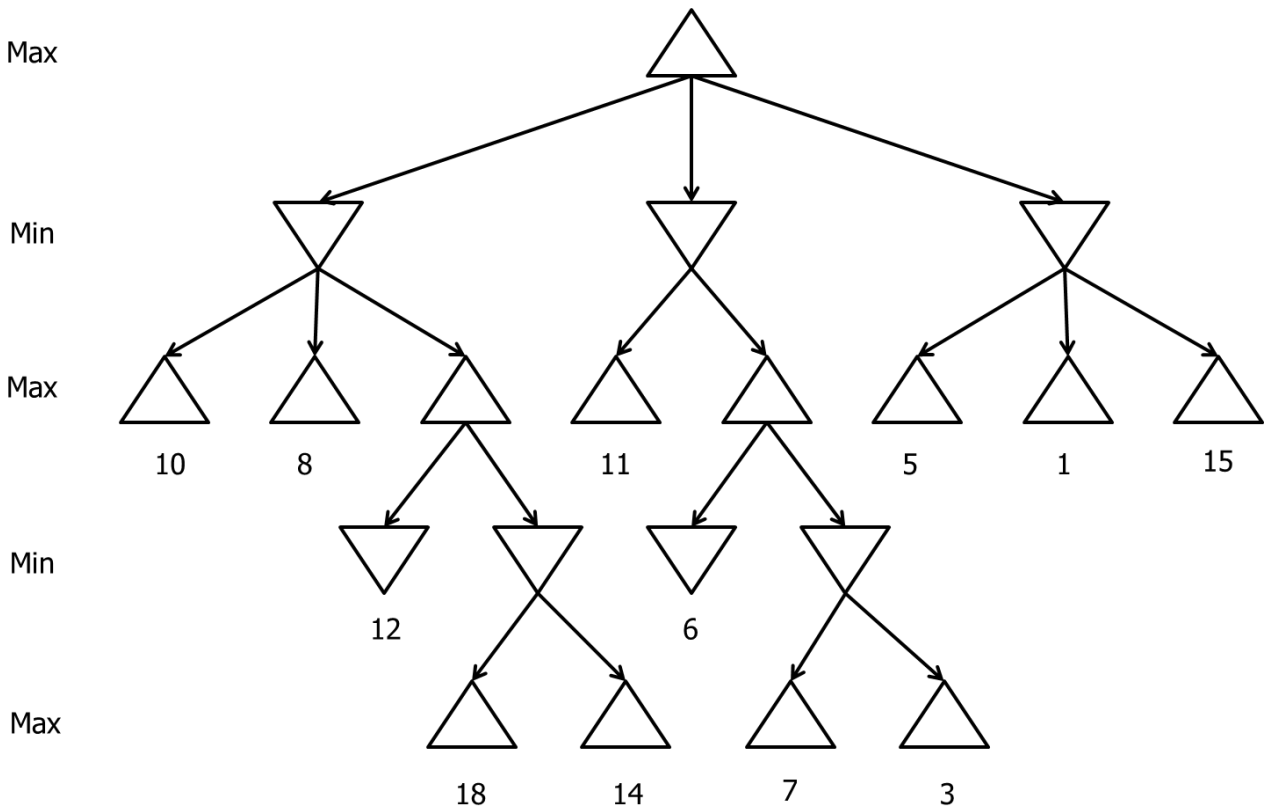**CS480 – Assignment 3**
Assigned: Monday, 9/23/2019
Due: 9:59pm (Chicago) on Sunday, 9/29/2019
There are 4 questions. Please submit your solutions through blackboard assignment page.

Use the following game tree for the first two questions.

1. Hand trace the mini-max algorithm. What is the maximum utility that MAX can achieve, assuming MIN plays optimally?
2. Hand trace the alpha-beta search. Show the updated bounds on the nodes. Clearly mark which branches are pruned, if any.

3. We are given the following CSP problem.

The variables and domains are as follows.
A: {4, 5, 6, 7, 8}
B: {10, 20, 30, 40}
C: {2, 3, 4}
D: {28, 43, 56, 77, 94, 114}

The constraints are:
A + C is odd.
A + D is a square of an integer.
B + D < 60.

Solve this problem using the following heuristics and algorithms.

- Use backtracking search.
- For variable ordering, use MRV. If there are ties, use degree to break them. If there are still ties, break them in alphabetical order.
- For value ordering, order values from smallest to largest.
- For inference, use forward checking.

Please show the search tree and the stack of the domains (see the lectures for examples). When the search backtracks due to an empty domain, show it clearly on your search tree. Write down the final solution, if there is any.

4. This is the same cryptarithmetic problem we solved in class, except the domains of the variables are different.

TWO + TWO = FOUR

Assume that F is already assigned 1. In this version, we'll assume that there will be a carry over from O + O and there will be a carry over from W + W. Hence, the constraints are now:

O + O = 10 + R
W + W + 1 = 10 + U
T + T + 1 = 10 + O
And, of course, all digits are distinct. Assume the following domains:

O: {6, 7, 8, 9}
R: {0, 2…9}
W: {5…9}
U: {0, 2…9}
T: {5…9}

Solve this problem using the following heuristics and algorithms.

- Use backtracking search.
- For variable ordering, use MRV. If there are ties, break them in the following order: O, R, W, U, T.
- For value ordering, order values from smallest to largest.
- For inference, use forward checking.

Please show the search tree and the stack of the domains (see the lectures for examples). When the search backtracks due to an empty domain, show it clearly on your search tree. Write down the final solution, if there is any.