Documentation: AI Programming Assignment 1

Game Playing Strategies

Usage:
Go to the Python Shell and type the following and press Enter
>>> python <filename.py>

Result:

Minimax Strategy:
We employ a minimax game playing strategy where we

Alpha-Beta Pruning:
This method is applied where we want to reduce the no. of nodes traversed in a tree, and save a lot of running time when it comes to making decisions for players (of a game). To do this, we first use the utility values of the leaf nodes, and work our way upwards to find out the minimax values of each node of the tree. Based on that we do two tests:

For a Min state node:

Case1: If the child vertices are lesser than the minimax (Beta) value of the node AND the child vertices are greater than the current Max value of the super-ancestor node (parent of the parent node), then we copy the value of the child onto the parent.

Case2: If the child vertices are greater than the minimax value of the node, then we move on to the next child vertex

Case3: If the child vertices are greater than the minimax value of the parent node, BUT they are lesser than the Max value of the super-ancestor; It means that no other child vertex can be preferred to be copied onto the parent. Hence we PRUNE the other nodes, and move on to another sub-tree.

The same logic can be conversely applied for a Max state node having a minimax value(Alpha).

In this way we, we are able to ignore a lot of unnecessary nodes and traverse the entire tree in a faster way to get the (Alpha, Beta) values of the root node. Then we can backtrack our path based on the same values which got us to the root node, and get the optimal game playing decision sets for a particular side.

Below is the output of the minimax tree with Alpha – Beta pruning

```
Input tree_value is:
(
5

(
(
(
4

7

-
2
)

7
)

6
)
)

Pruning will occur for the MIN value of  4 at position 4
7
-2
)

The tree has the following Alpha and Beta Values: 6
Solution path is:
2
2
```