# DA221: Introduction to AI

## Lab Assignment 2: TIC TAC TOE SOLVER

Priyansh Awasthi
220150013

## ALGORITHMS USED

1. MINIMAX without alpha beta Pruning
2. MINIMAX with alpha beta Pruning or NORMAL

## METRICES FOR EVALUATION:

1. Number of Nodes Visited
2. Time taken

## ANALYSIS:

### LANGUAGE USED : C++

### NUMBER OF NODES VISITED:

When both algorithms were placed to fight each other in Tic Tac Toe game with randomly selecting which algorithm starts first, these are the number of nodes both algorithms visited in their respective MINIMAX functions.

| Seed | Number of Instances | Nodes Visited(Pruning) | Nodes Visted (Normal) |
|---|---|---|---|
| 100 | 100 | 1204572 | 3182202 |
| 101 | 100 | 1129965 | 3292784 |
| 102 | 100 | 1122548 | 3411420 |
| 103 | 100 | 1037970 | 3628846 |
| 104 | 100 | 1098933 | 3431190 |
| 105 | 100 | 1055091 | 3739324 |
| 106 | 100 | 1155328 | 3301470 |
| 107 | 100 | 1086190 | 3570246 |
| 108 | 100 | 1076105 | 3674104 |
| 109 | 100 | 1129463 | 3263240 |
| 110 | 100 | 1087462 | 3410414 |

Hence, on an average:

| Average Nodes Visited(Pruning) per match | Average Nodes Visited(Normal) per match |
|---|---|
| 11,076 | 34,459 |

It can be seen clearly from the above average number of nodes visited in 100 matches that MINIMAX algorithm with alpha-beta pruning is about 3 times efficient (visits $\approx$ 67% less nodes).

### TIME TAKEN (in microsecond):

To measure speed of solving the problem , i.e Time Taken to finish a match(either win/draw) , we measured the time taken by 100 matches between algorithms when:
1.Both were not using alpha beta pruning (Normal vs Normal)
2.One of them was using alpha beta pruining (Normal vs Pruning)

**VALIDITY**: This measurement metric is valid becuase during comaparing the performance between 2 types of matches, we have kept the seed value to same hence in both kind of matches *Normal Algorithm* plays the same against Pruning Algorithm and another Normal Algorithm.

| Seed | Number of Instances | Normal vs Normal | Normal vs Pruning |
|---|---|---|---|
| 100 | 100 | 323353 | 216902 |
| 101 | 100 | 303439 | 196324 |
| 102 | 100 | 311305 | 202262 |
| 103 | 100 | 288173 | 215007 |
| 104 | 100 | 297898 | 196959 |
| 105 | 100 | 282935 | 212311 |
| 106 | 100 | 295513 | 204645 |
| 107 | 100 | 287634 | 208778 |
| 108 | 100 | 290621 | 220713 |
| 109 | 100 | 290771 | 202619 |
| 110 | 100 | 295253 | 214375 |

Hence, on an average: (Time taken is in **microseconds**)

| Average Time Taken(Normal) per match | Average Time taken(Pruning) per match |
|---|---|
| 2969 | 2082 |

It can be seen clearly from the above average time taken for 100 matches that MINIMAX algorithm with alpha-beta pruning is about 1.5 times efficient (takes $\approx$ 33% less time).

## CONCLUSION:

Using both the metrices, we have shown that MINIMAX Algorithm with $\alpha$ $\beta$ Pruning is almost 1.5x better in terms of time taken and almost 3x better in terms of space complexity.
Thereotically as well, we can easily deduce that by comparing values of $\alpha$ and $\beta$ at each node, MINIMAX with Pruning should visit lesser nodes and hence should have lesser time and space complexity.
Hence, it can be concluded that MINIMAX with Pruning is far better than without pruning in case of **Two-Player, Zero-Sum game**.