

Play nine (9) human versus computer (using both algorithms) games, each starting with a different move. Count the total number of expanded nodes (sum of expanded nodes for every computer move) and report them in the table below.

Your (X) First move	Computer (O) with MiniMax algorithm. Total (for every move) number of generated nodes	Computer (O) with MiniMax with alpha beta pruning algorithm. Total (for every move) number of generated nodes
1	59705, 1053, 47, 5	2339, 76, 18, 6
2	63905, 1055, 45, 4	2870, 270, 32, 5
3	59705, 1053, 47, 5	3276, 366, 43, 6
4	63905, 1229, 45, 4	3575, 180, 28, 5
5	55505, 1055, 45, 4	2317, 280, 43, 6
6	63905, 1055, 45, 4	3591, 394, 44, 6
7	59705, 1053, 47, 5	3810, 145, 41, 6
8	63905, 1349, 49, 5	4982, 190, 34, 6
9	59705, 1053, 47, 5	3958, 319, 43, 6

What are your conclusions? Which algorithm performed better? Write a short summary below.

Both programs acted the same in terms of winning vs losing but alpha beta pruning clearly performed better in terms of runtime. Comparatively, it went from an average of 60,000 nodes on the first tree to 3000. That is nearly 1/20th of the nodes just from pruning. The code change to account for alpha beta pruning also only added one extra check meaning this is far more efficient even with extra logic. Even with such a simple game such as tic tac toe, simple min max sometimes took a bit too long to run. Alpha beta pruning is clearly preferred.