

- 1. Compare the Minimax values for A4Q1 and A4Q2. The minimax value used a different scale (-1,1) versus a count of the number of queens, but did this change which player won the game?**

The different variations didn't change which player won the game. The only different is variation1a just shows which player won at the end, variation1b also shows how many queens on the board.

- 2. Compare the runtimes for A4Q1 and A4Q2. Which variation seemed to be faster? Explain why.**

The runtimes for A4Q1 and A4Q2 are similar, because the only different between this two variations is the returning from utility() function. The depth, the nodes traversed are same.

- 3. Compare the Minimax values for A4Q1 and A4Q3. Were they the same?**

Yes, the Minimax values from A4Q1 and A4Q3 are same.

- 4. Compare the runtimes for A4Q1 and A4Q3. How much faster was the search, for games with larger N?**

The A4Q3's runtime much faster than A4Q1's runtime with the exponential growth rate. And the number of faster will be much bigger with the increasing of the size of the game board. Specifically, the runtimes from A4Q1 in 5 * 5 board is 0.00158 seconds, the runtimes from A4Q3 in 5 * 5 board is 0.000523 seconds. But the runtimes from A4Q1 in 12 * 12 board is 1.39 seconds, the runtimes from A4Q3 in 12 * 12 board is 0.126 seconds.

- 5. Compare the Minimax values for A4Q2 and A4Q4. Were they the same?**

Yes, the Minimax values from A4Q2 and A4Q4 are same.

- 6. Compare the runtimes for A4Q2 and A4Q4. How much faster was the search, for games with larger N?**

The A4Q4's runtime much faster than A4Q2's runtime with the exponential growth rate. And the number of faster will be much bigger with the increasing of the size of the game board. Specifically, the runtimes from A4Q2 in 5 * 5 board is 0.00133 seconds,

the runtimes from A4Q4 in $5 * 5$ board is 0.000563 seconds. But the runtimes from A4Q2 in $12 * 12$ board is 1.39 seconds, the runtimes from A4Q4 in $12 * 12$ board is 0.144 seconds.

7. Compare the Minimax values for A4Q1 and A4Q5. Were they the same? Explain.

No, the Minimax values from A4Q2 and A4Q4 are different. Because the A4Q5 has the limitation of the depth, so each minimax value from A4Q5 may not optimal. The A4Q2 doesn't has the limitation of the depth, so the minimax value from A4Q2 is optimal.

8. Compare the runtimes for A4Q1 and A4Q5. How much faster was the search, for games with larger N?

The A4Q5's runtime much faster than A4Q1's runtime, and after size 5 of the board, the number of faster growing with the exponential growth rate. Specifically, the runtimes from A4Q1 in $5 * 5$ board is 0.00153 seconds, the runtimes from A4Q3 in $5 * 5$ board is 0.00110 seconds, it almost same speed. But the runtimes from A4Q1 in $12 * 12$ board is 1.39 seconds, the runtimes from A4Q3 in $12 * 12$ board is 0.021 seconds.

9. Compare the Minimax values for A4Q5 and A4Q6 (Minimax with cut-off, two variations) with the results from A4Q7 (Alpha-Beta with cut-off, two variations). Were they the same? Explain.

No, the Minimax values are different from A4Q5 and A4Q6 with the A4Q7. Because A4Q5 and A4Q6 check all the nodes in game tree, then there are more possible not optimal value from each node. But the A4Q7 use depth cut off with alpha-beta, there will be less wrong or bad value from each parent nodes, but it also may result to cut the wrong paths because wrong value from children nodes.

10. Compare the runtimes for A4Q5 and A4Q6 (Minimax with cut-off, two variations) with the results from A4Q7 (Alpha-Beta with cut-off, two variations). How much faster was the search, for games with larger N?

The A4Q7's runtime much faster than A4Q5 and A4Q6's runtime, and it approximately 10 times faster for each script.