**Question:  
N\_Queens Problem:**

**Comparative Analysis of Local Search Algorithms for the 8-Queens Problem**

In this analysis, we compare Hill Climbing, Simulated Annealing, and Local Beam Search based on their performance in solving the 8-Queens Problem.

1. **Effectiveness Comparison:**

Each algorithm attempts to minimize queen conflicts while searching for an optimal solution.

* **Hill Climbing** is efficient but often traps in local minima and may not always find a solution.
* **Simulated Annealing** introduces randomness, allowing it to escape local optima, making it more effective.
* **Local Beam Search**, by considering multiple candidate states, has the highest success rate but requires more computational resources.

**2. Fastest Method to Find a Solution**

Hill Climbing is usually the fastest when a solution is easily reachable. However, if it gets stuck in a local minimum, it may never find a solution without restarts.

Simulated Annealing is slower but more robust, as it gradually refines solutions without getting stuck.

Local Beam Search can be the fastest when multiple paths are explored efficiently, but if computational resources are constrained, it might be slower than Simulated Annealing.

Overall, **Local Beam Search is often the most efficient in reaching a solution quickly while ensuring success**.

**3. Impact of Randomness in Simulated Annealing**

Simulated Annealing uses a temperature function that gradually decreases over time. The randomness helps in:

Escaping local optima by allowing worse moves early on.

Exploring a diverse set of solutions before converging.

Reducing premature convergence, unlike Hill Climbing.

However, if the temperature function is not well-tuned, it may either:

Explore too much, taking longer to find a solution.

Converge too quickly, behaving similarly to Hill Climbing.

**4. Role of Visualization in Understanding Search Behavior**

Visualization helps in:

Observing step-by-step queen movement to see how conflicts are resolved.

Identifying local optima and plateaus, particularly in Hill Climbing.

Understanding how randomness (Simulated Annealing) and multi-state search (Local Beam Search) influence progress.

Comparing the effectiveness of different algorithms visually.

**5. Which Algorithm Escapes Local Optima the Best?**

Simulated Annealing is the best at escaping local optima because:

It accepts worse solutions probabilistically early on.

The decreasing temperature allows gradual refinement.

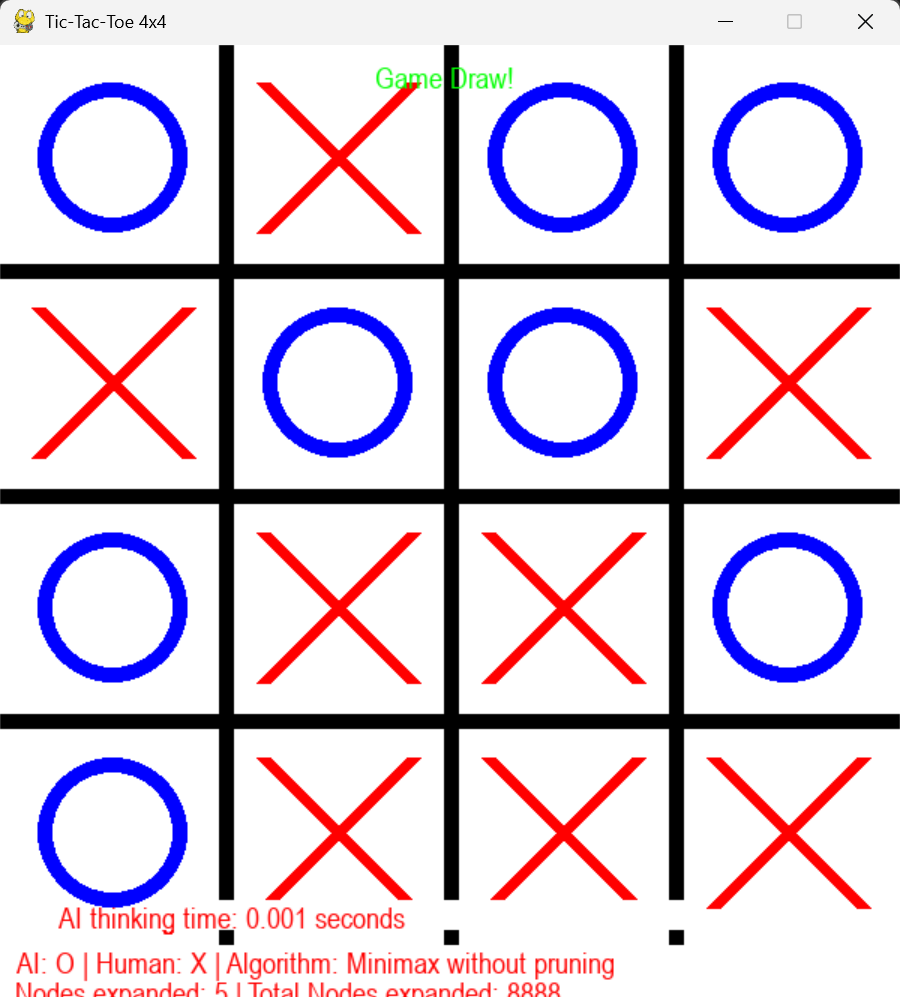
Local Beam Search also avoids local optima but requires tracking multiple states, which increases complexity.

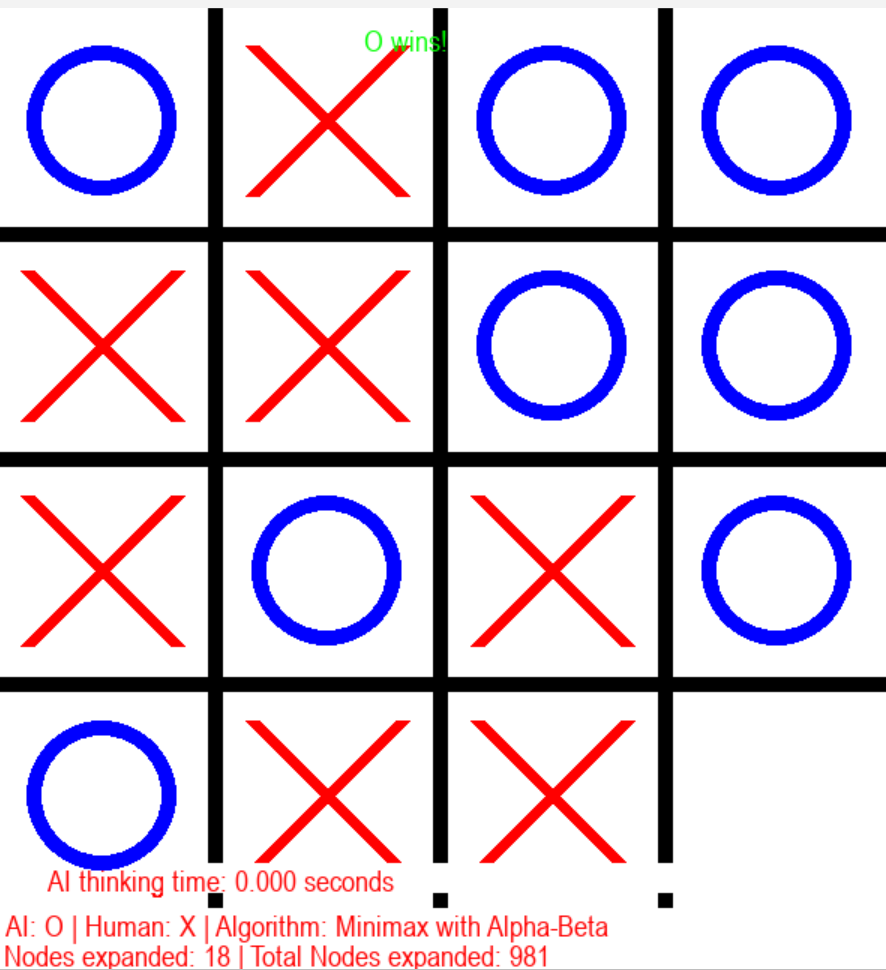
**Hill Climbing is the worst at escaping local optima as it only accepts improvements.**

**Question:  
Tick\_Tack\_Toe:**

**1)**

**Without pruning Total Nodes expanded=8888**

**With pruning Total Nodes expanded=981**

**With Alpha\_beta Pruning:**

Alpha-Beta Pruning performed significantly better in terms of speed. The total nodes expanded were reduced from 8888 to 981, which shows a significant optimization. Execution time was also slightly better, making the AI's move decision faster.

**2. Which method performed better in terms of speed?**

Alpha-Beta Pruning performed significantly better in terms of speed. The total nodes expanded were reduced from 8888 to 981, which shows a significant optimization. Execution time was also slightly better, making the AI's move decision faster.

**3. How much reduction in node expansion did Alpha-Beta Pruning provide?**

By comparing nodes expanded:

**Without Pruning:** 8888 nodes

**With Pruning:** 981 nodes

**Reduction in Nodes Expanded =** 8888−981=7907

**Percentage Reduction =** (7907/8888)×100=89% reduction

Alpha-Beta Pruning dramatically reduces the number of nodes that need to be evaluated, making it more efficient.

**4.Does playing first or second affect AI performance? Why?**

Yes, the order of play can impact AI performance

If **AI** **plays first**, it has more control over the game and can dictate the strategy.

If **AI** **plays second**, it has to respond to the human player’s moves, which may limit its ability to implement an optimal strategy immediately.

Since Minimax explores all possibilities, the first player typically has a slight advantage because they can steer the game in a more controlled manner. However, with Alpha-Beta Pruning, this effect is minimized because the AI efficiently prunes unnecessary branches, allowing it to react effectively even when playing second.

**Question:**

**1. Hill Climbing with Restarts**

* **Speed:**
  + Very fast move selection (typically 0.003–0.009 seconds per move).
* **Search Characteristics:**
  + Uses only 5–10 iterations per move.
  + The multiple restarts help in escaping some local optima, but the overall greedy nature means that it sometimes converges to moves with poor evaluations (even highly negative scores later on).
* **Trade-off:**
  + While it delivers moves quickly, the method may not consistently find the best move because it doesn't fully explore deeper opponent responses.
  + Its performance is acceptable for shallow searches or time-critical decisions but can be strategically suboptimal.

**2. Simulated Annealing**

* **Speed:**
  + Slightly slower per move (around 0.016–0.035 seconds), because it runs a fixed number of iterations (here, 100) regardless of early improvements.
* **Search Characteristics:**
  + The algorithm accepts worse moves probabilistically (based on a temperature schedule) to help escape local maxima.
  + This extra exploration sometimes leads to better moves initially, but later in the game, it also sometimes produces very low scores.
* **Trade-off:**
  + It tends to explore more of the search space than hill climbing, which might yield more robust moves in complex positions, albeit at the cost of a longer computation time per move.

**3. Minimax with Alpha-Beta Pruning**

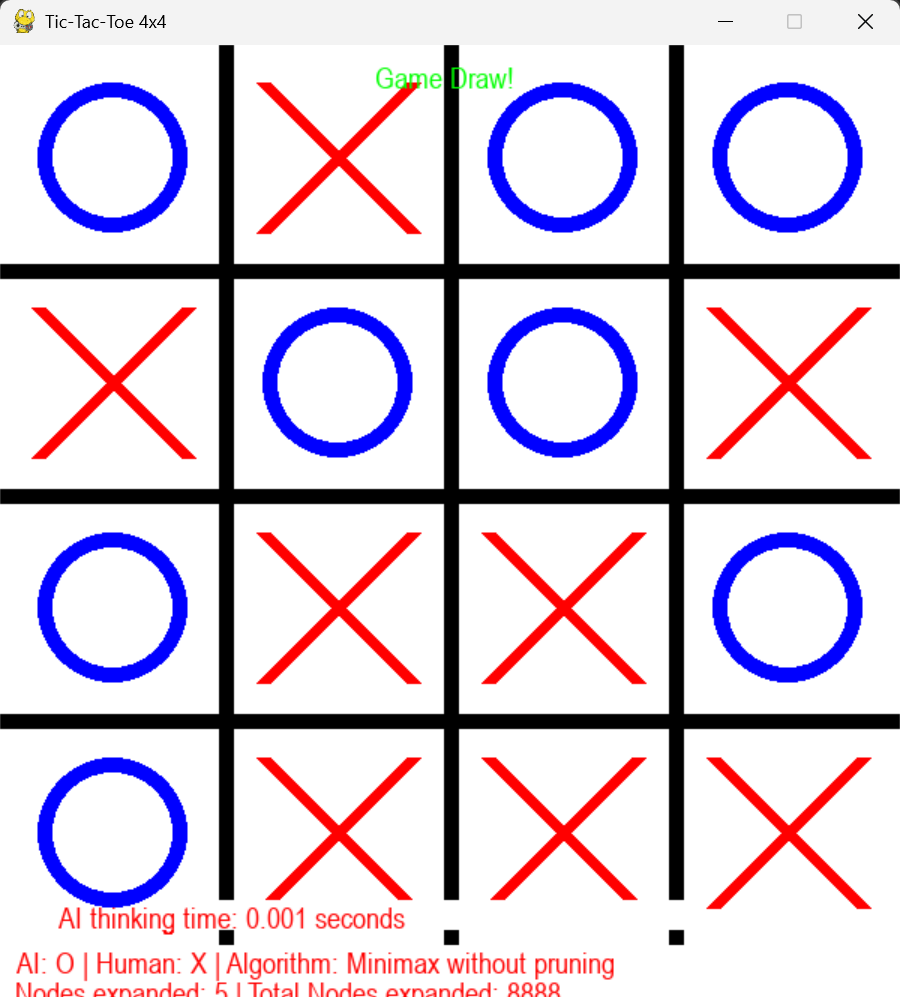
* **Speed:**
  + Moderately fast, with move times ranging from about 0.008 to 0.06 seconds.
  + The node count varies between roughly 110 and 480 nodes per move depending on the board state.
* **Search Characteristics:**
  + Unlike the local search methods, minimax with alpha-beta considers both the AI's and the opponent’s moves, leading to a more comprehensive evaluation.
  + This results in a more robust decision-making process.
* **Trade-off:**
  + While minimax tends to be a bit slower than hill climbing, its consideration of opponent moves gives it a significant strategic advantage.

**Overall Comparison**

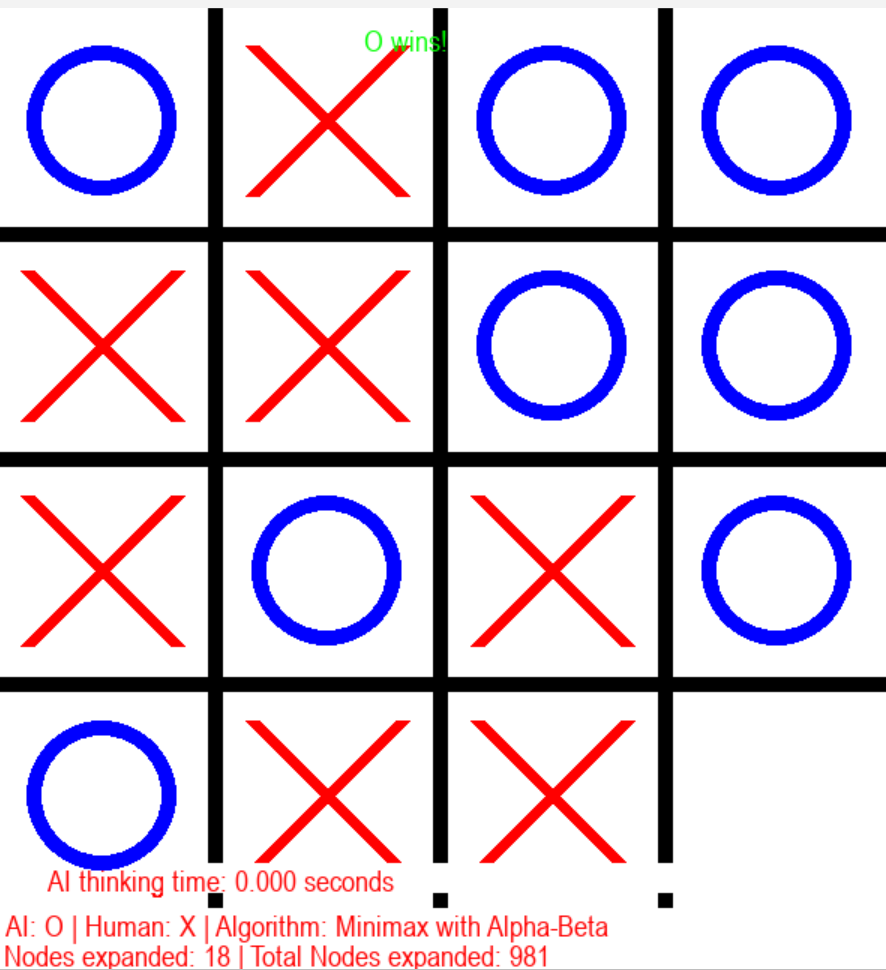
* **Efficiency vs. Quality:**
  + **Hill Climbing** is extremely efficient in terms of time but can get trapped in local optima, sometimes resulting in poor move choices.
  + **Simulated Annealing** offers improved exploration with a controlled trade-off in speed, making it more likely to escape poor local maxima, though it still has its limits.
  + **Minimax with Alpha-Beta** offers a more complete search (by considering both players) and generally produces more robust moves, though it comes at a moderate computational cost.

Outputs:

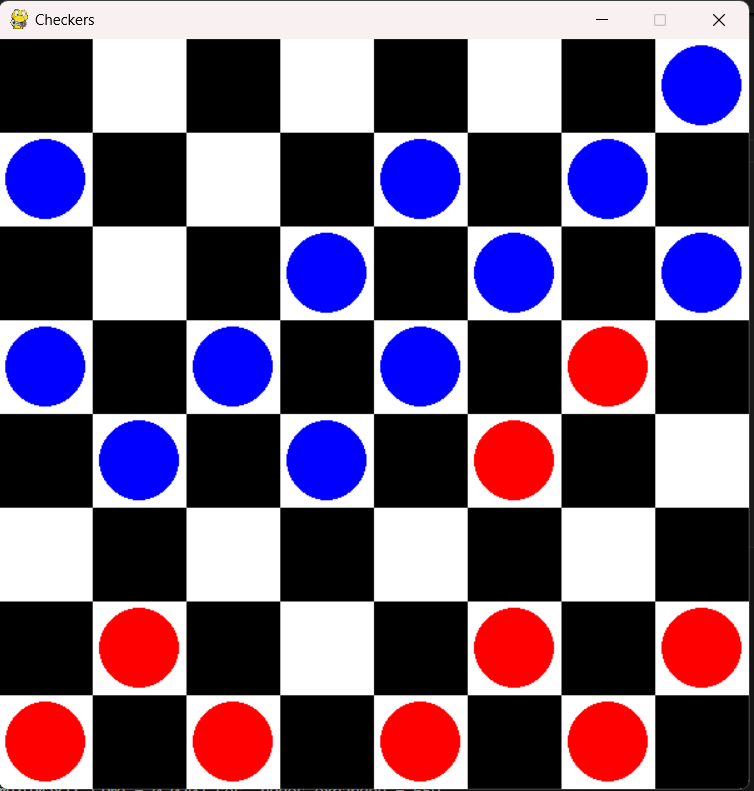
TICTAC TOE

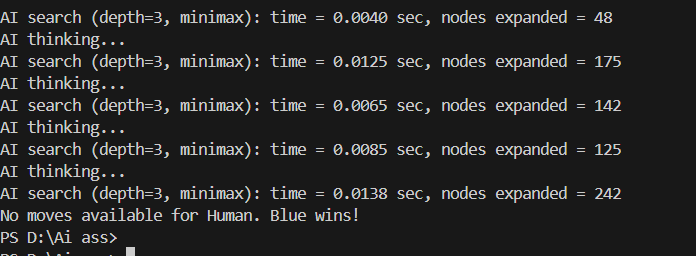


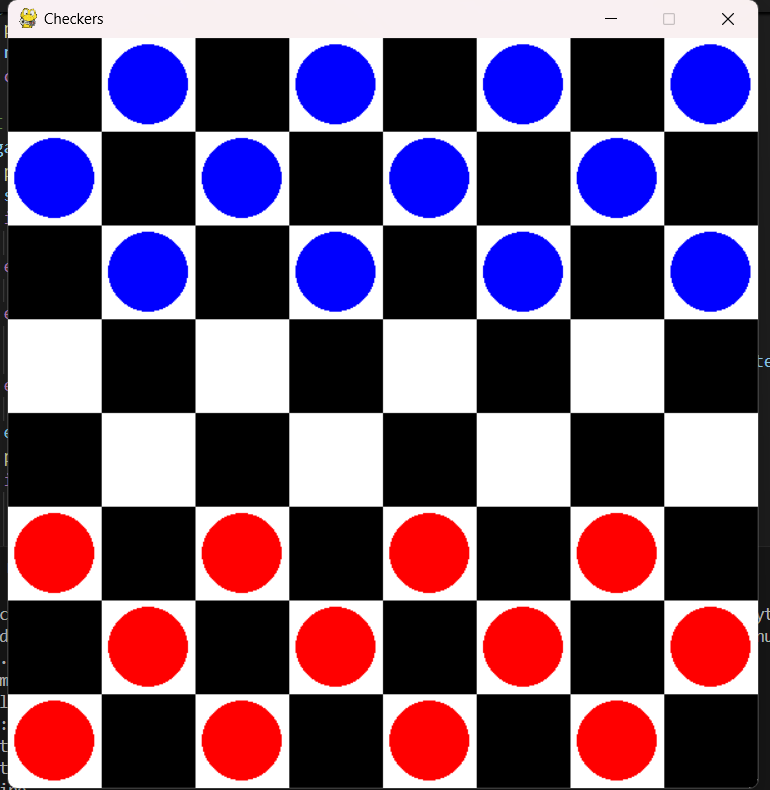
**With apha beta pruning**

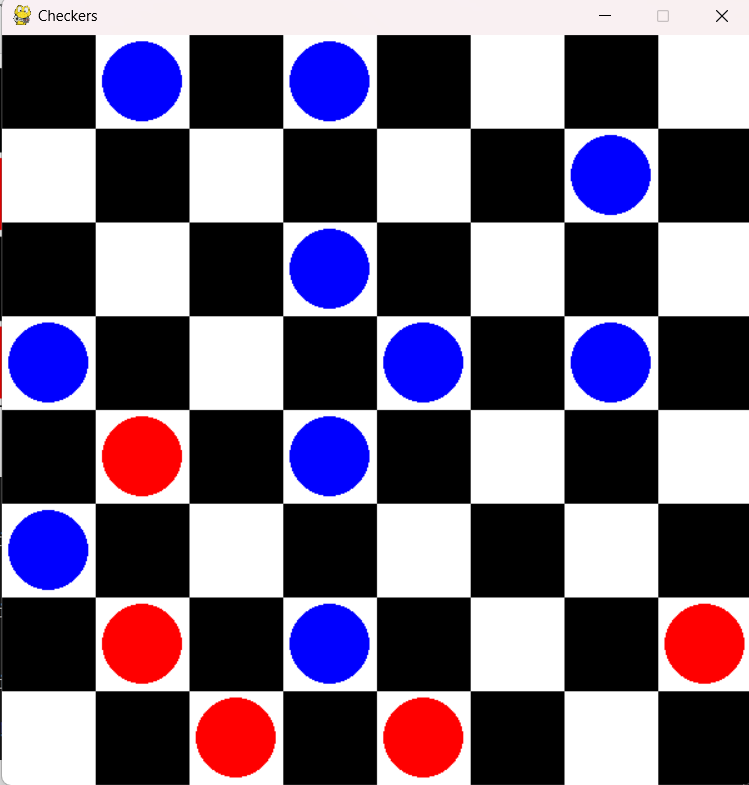


Checkers:

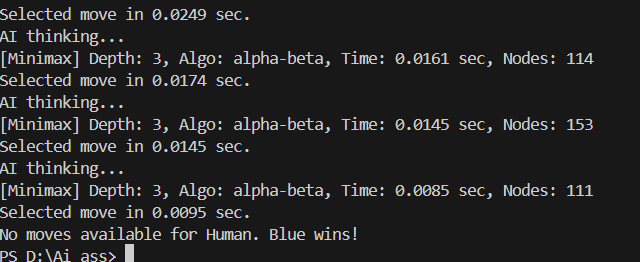


  
**Checkers BONUS:**

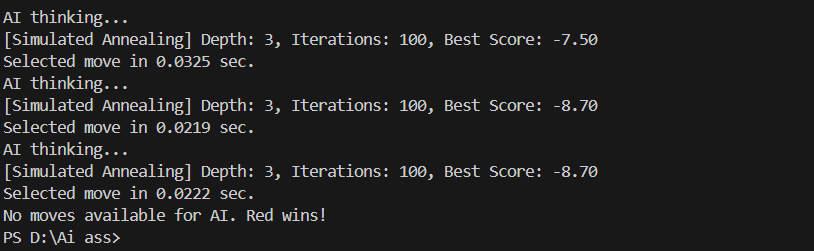




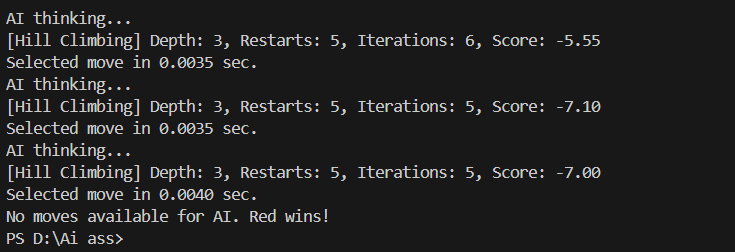
Minmax



Simulated annealing



Hill Climbing



N\_QUEENS\_Problem:

Innitial\_State:  
