Minesweeper Final AI Report

Team number_____8_____
Member #1 (name/UCI netid)_____Douzi Ma/10161248_____
Member #2 (name/UCI netid)____Serena Nguyen/64283133_____

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm. What did you do that was fun, clever or creative?

Our Minimal AI algorithm for Minesweeper mainly uses the strategy from the lecture slide. First, it looks for safe moves by finding uncovered '0' tiles and picking an adjacent covered tile. If that doesn't work, it gets a bit fancier. It identifies the "frontier" and uses a technique called model checking. This involves trying out all possible mine configurations for the frontier and seeing which ones fit with the numbers we can see.
What's neat about this approach is how it combines straightforward strategies with more advanced reasoning. The model checking part lets the AI make educated guesses when there's no obvious safe move. Plus, the way it's set up makes it easy to add new strategies later. If all else fails, it falls back to picking the tile closest to the center, which is a simple but effective last resort. Overall, it's a fun mix of simple and smart that can handle 100% situations in the 5x5 super easy level game.

I.B Describe your Minimal AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5	**1000**	**1000**	**100%**
8x8	3	3	0.03%
16x16	0	0	0%
16x30	N/A	N/A	N/A
Total Summary	1000	1000	100%

II. Final AI

II.A. Briefly describe your Final AI algorithm, focusing mainly on the changes since Minimal AI:

We decided to make big changes Minesweeper AI algorithm because the Minimal AI, while capable of solving super easy level games, struggled with beginner, intermediate and expert levels in terms of both time efficiency and win rate.

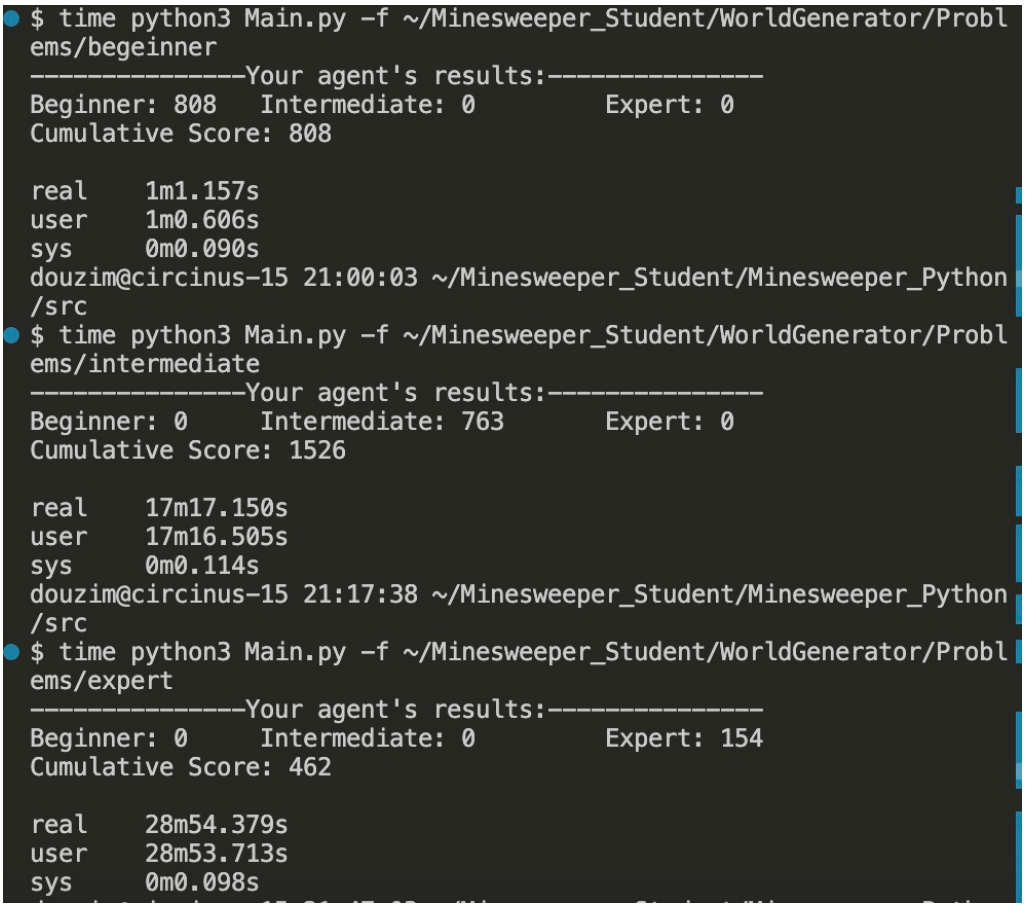
The new algorithm addresses these limitations by introducing a more accurate board representation and decision-making process. It now stores more detailed information about each tile, including effective labels and adjacent covered tile counts, allowing for more nuanced reasoning.

We've added a queue-based system for managing moves, implemented a board-wide processing approach after each action, and incorporated advanced techniques like the subset neighbor algorithm, it can uncover information about tiles that aren't directly adjacent to any single numbered tile, but are instead caught between two or more numbered tiles. It's a key reason why your updated AI can handle more complex Minesweeper scenarios that the Minimal AI couldn't solve efficiently. Here is how it works:

1. For each numbered tile on the board, the algorithm looks at its neighboring tiles.
2. It then compares the set of covered neighbors of this tile (let's call it Tile A) with the set of covered neighbors of each of its own numbered neighbors (let's call one of these Tile B).
3. If Tile B's set of covered neighbors is a subset of Tile A's covered neighbors, the algorithm can make some powerful deductions:
    a. If the difference between Tile A's and Tile B's effective values (number minus adjacent flagged mines) is zero, then all the exclusive neighbors (those in A but not in B) must be safe.
    b. If the difference between the effective values equals the difference in the number of covered neighbors, then all the exclusive neighbors must be mines.

These changes enable the AI to make more complex deductions, handle trickier scenarios, and make smarter choices when there's no obvious safe move. The goal was to create an AI that could perform well across all difficulty levels, improving both its problem-solving capabilities and overall efficiency in tackling more challenging Minesweeper configurations.

Our final algorithm achieved satisfactory scores and execution times across all difficulty levels:



II.B Describe your Final AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5	N/A	N/A	N/A
8x8	1000	808	80.8%
16x16	1000	1526	76.3%
16x30	1000	462	15.4%
Total Summary	3000	2796	57.5%

Our Final AI is a significant improvement of Minimal AI, in terms of both time efficiency and win rate.

III. In about 1/4 page of text or less, provide suggestions for improving the performance of your system.

While our AI demonstrates satisfactory performance on 8x8 and 16x16 boards, we recognize the need to enhance its winning rate on the more challenging 16x30 expert board. To achieve this, we propose two primary enhancements:

1. Depth-First Search (DFS) Implementation: We suggest implementing a DFS algorithm to explore all valid mine combinations on the frontier. This approach could lead to more informed decisions in complex situations, potentially identifying safe moves that our current logic might overlook. By analyzing these combinations, we could safely uncover tiles that remain unflagged across all valid scenarios, and conversely, flag tiles that are consistently identified as mines. However, we acknowledge that this method can be computationally expensive, especially on expert-level boards. To address this, we would need to implement careful optimizations to strike a balance between thoroughness and efficiency, aiming to boost our AI's success rate on larger boards without significantly impacting execution time.
2. Prioritized Move Queue System: In conjunction with the DFS approach, we propose enhancing our existing to_be_uncovered queue with a priority-based ordering system. This system would prioritize moves based on their potential to provide useful information or their likelihood of being safe. Higher priority would be assigned to tiles adjacent to low-numbered uncovered tiles and those that could potentially unlock large sections of the board. Implementing this system using a priority queue or sorted list, with dynamically updated priorities, could lead to faster board solving by addressing high-information moves first. This strategic ordering of moves is expected to improve decision-making in complex situations, potentially resulting in more efficient gameplay and higher success rates on challenging boards.

By implementing these enhancements, we aim to optimize our AI's performance on expert-level Minesweeper boards, improving its winning rate while maintaining efficient execution times. These additions build upon our existing algorithm, offering a balanced approach to tackling the complexities of larger Minesweeper boards.