Solving Minesweeper with AI

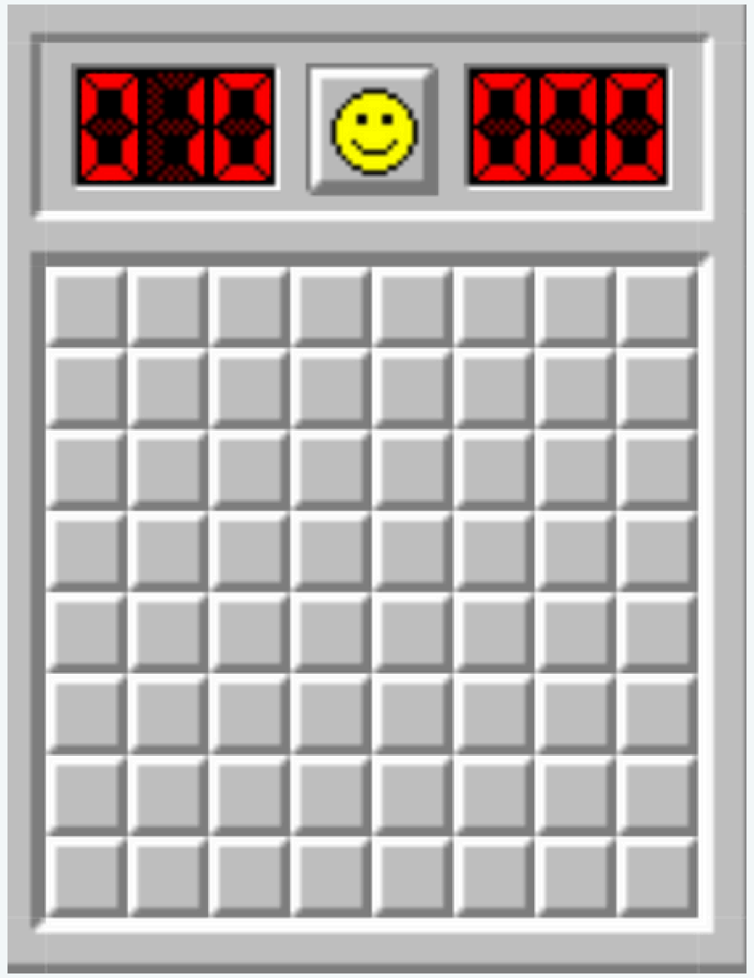
Students: Sona Stepanyan,
Anna Minasyan, Mane Koshkaryan,
Armen Nalbandyan
Instructor: Monika Stepanyan

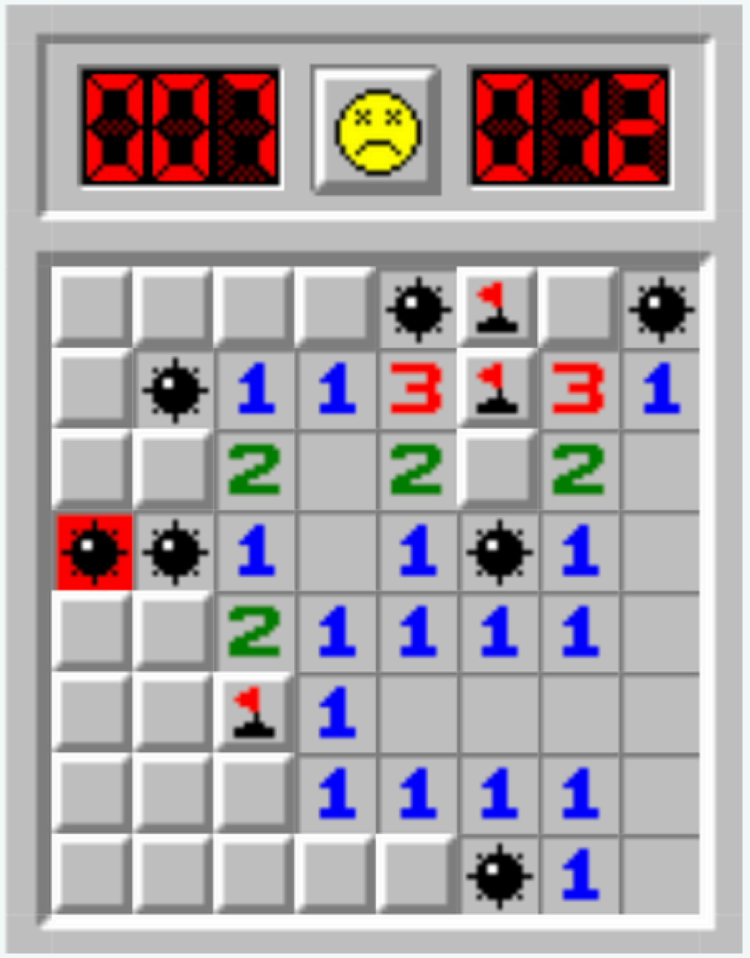
Today we will talk about

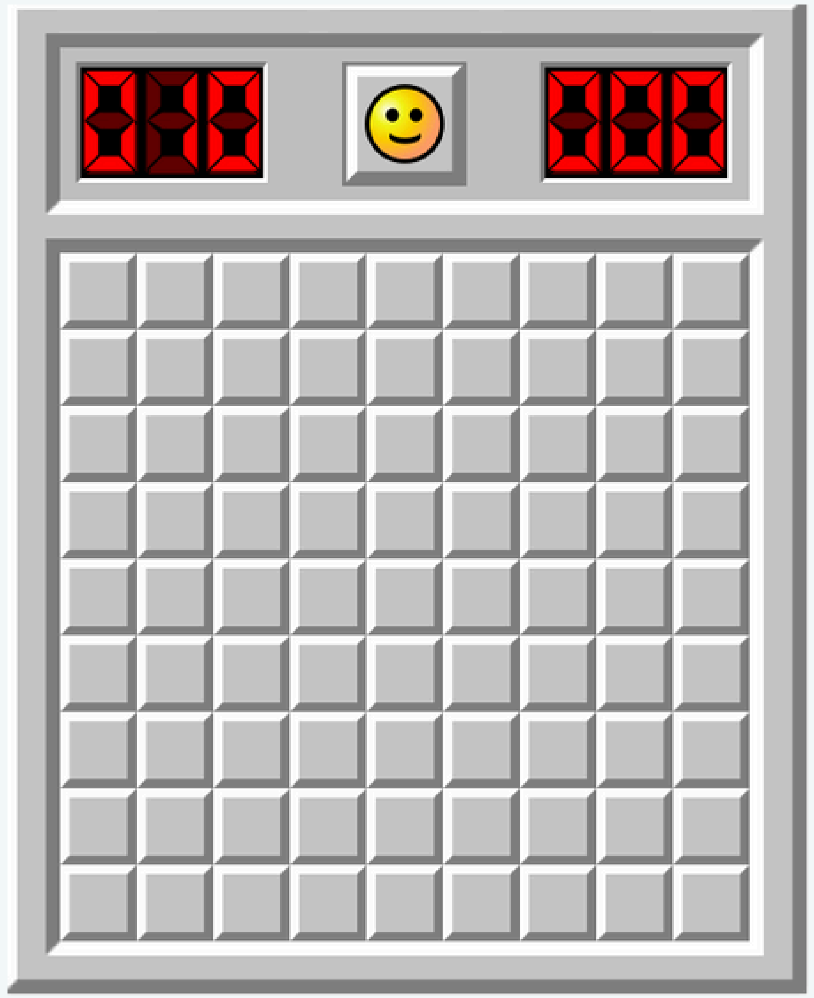
-Minesweeper
-Essential Considerations for Minesweeper
-DFS in Minesweeper
-CSP in Minesweeper
-Probabilities and Logic in Minesweeper
-CNF-SAT in Minesweeper
-Conclusion

Introduction

- Single-player puzzle game on an M×N grid.
- Non-deterministic environment.
- Player can uncover cells or flag suspected mines.
- Numbers (0–8) indicate adjacent mines; "0" uncovers connected safe cells.
- Objective: Uncover all safe cells and flag mines to win.
- Game ends with either a win or a loss based on player actions.
- Performance Measure: time taken to solve the puzzle.



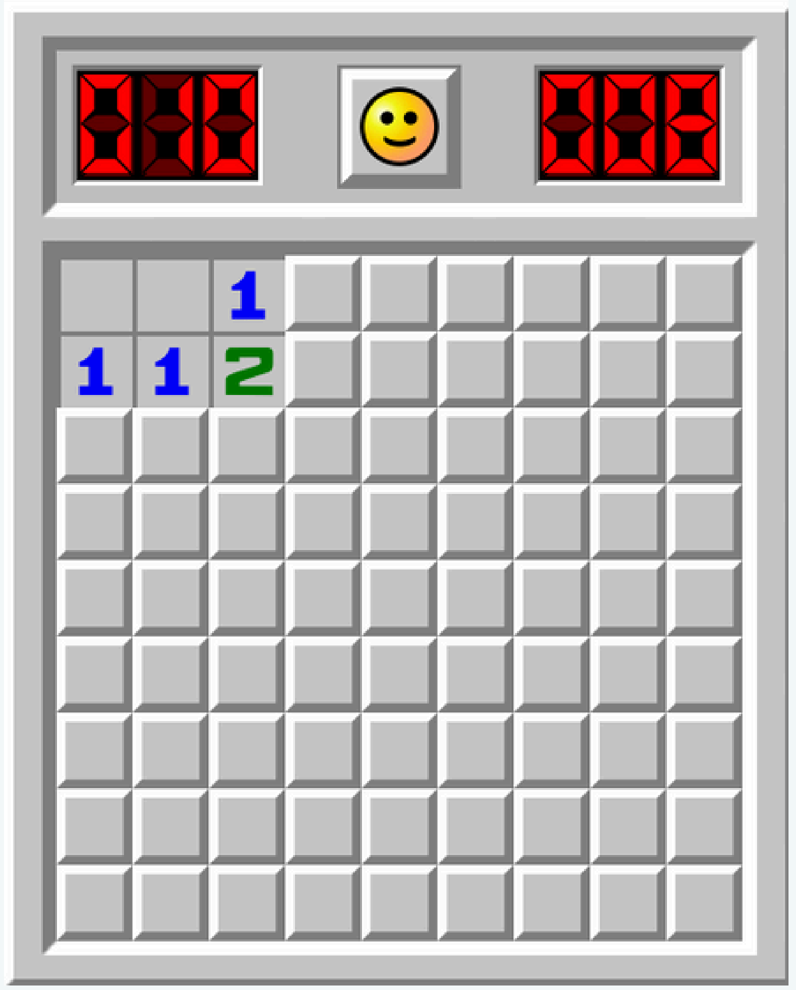




First Move

Always open the corner first

WHY

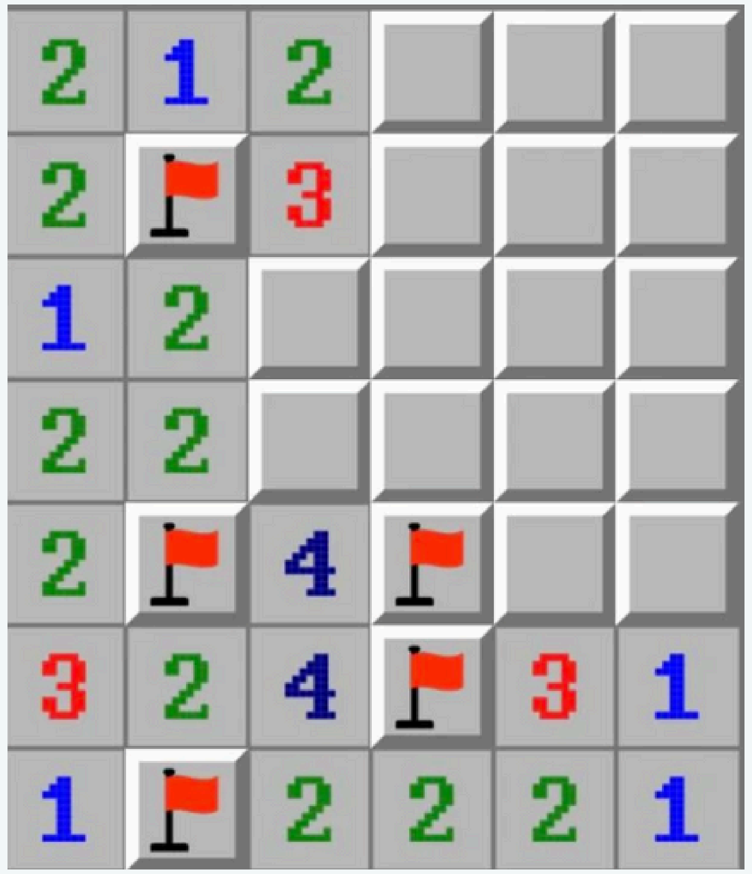


Advantages

The most likely to contain “0” after uncovering

Disadvantages

Reveals fewer adjacent cells and makeing the constraint limited



Dealing with Uncertainty in 50/50 Situations

Random Selection

WHY

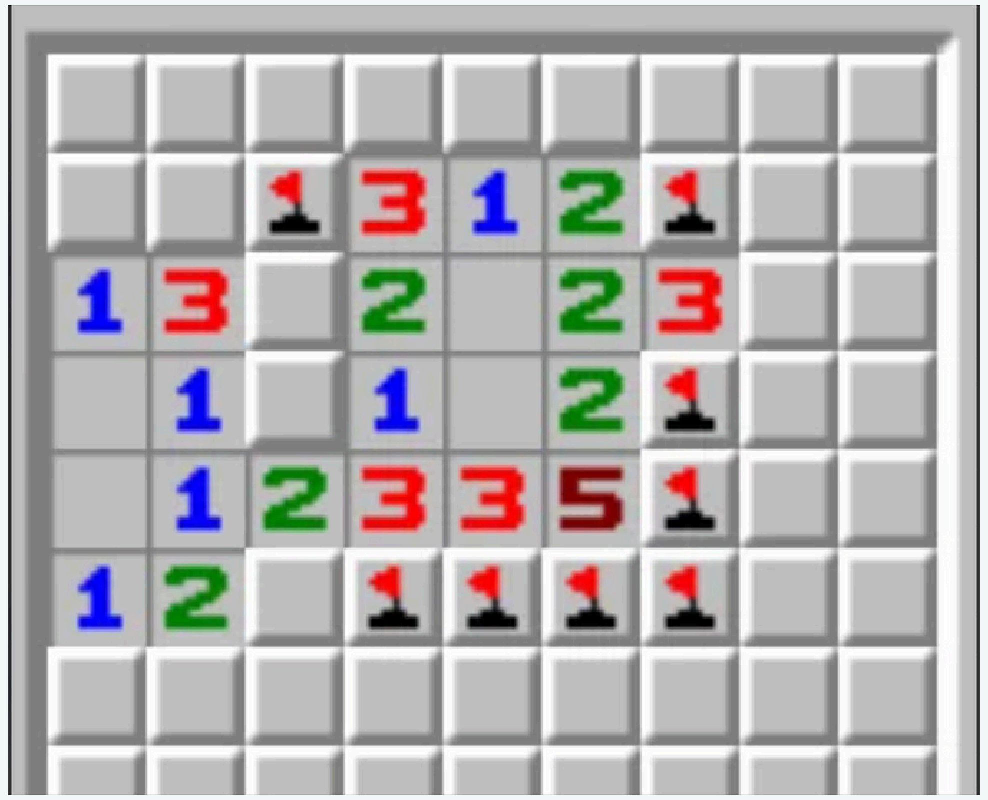
Advantages

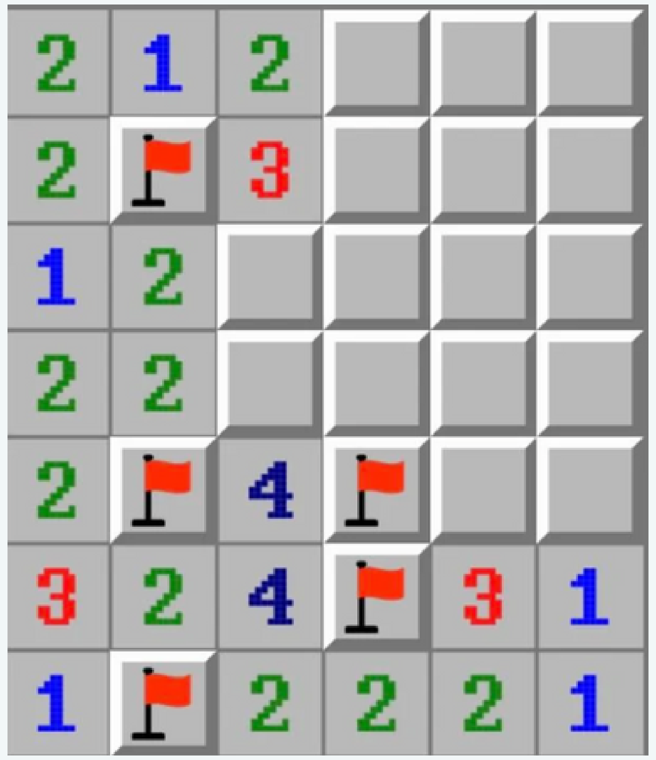
Easy to implement.
Ensures progress without getting stuck.

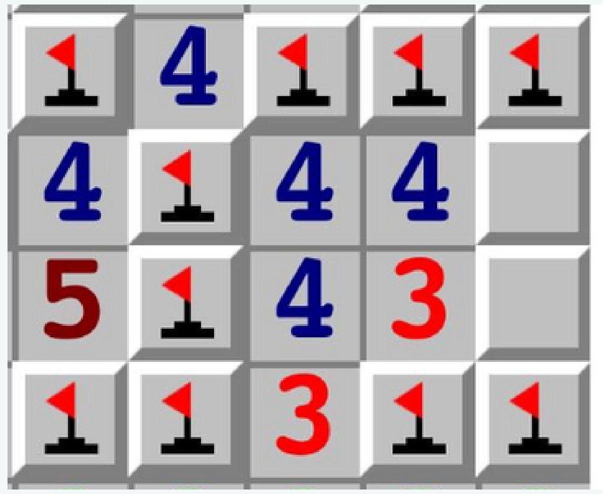
Disadvantages

Does not use board knowledge.
Struggles in high-difficulty levels with frequent guessing.

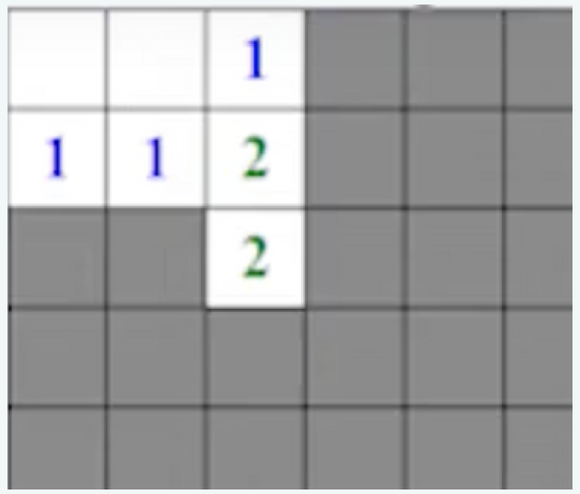
Probabilities and Logic in Minesweeper

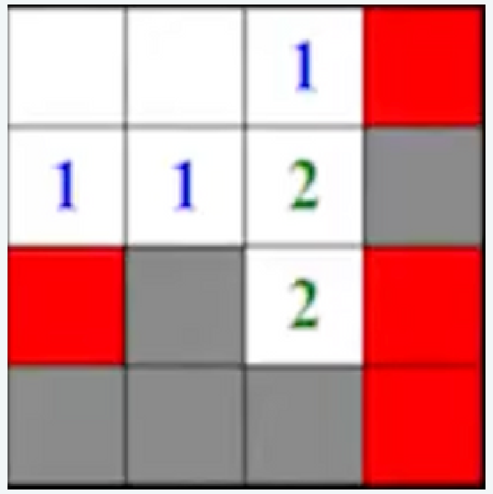




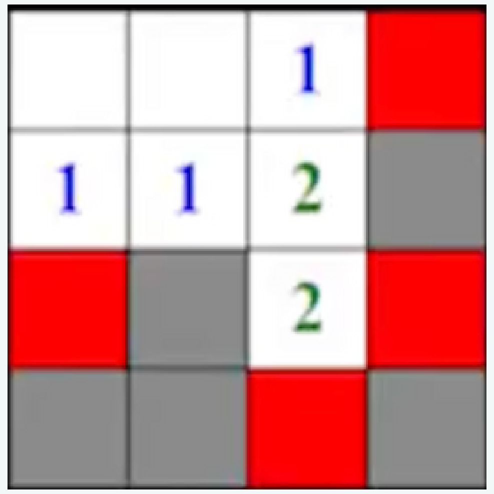


Probabilities and Logic in Minesweeper

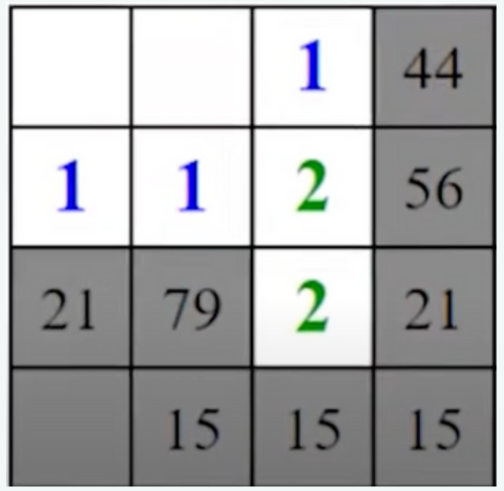


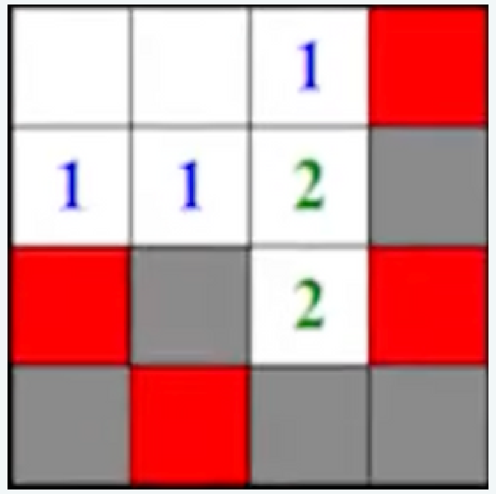


4 mines

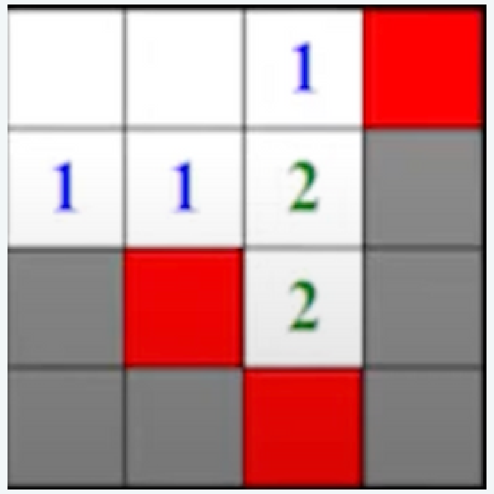


4 mines

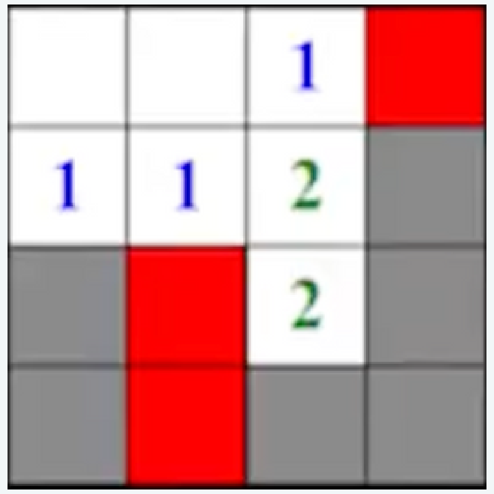




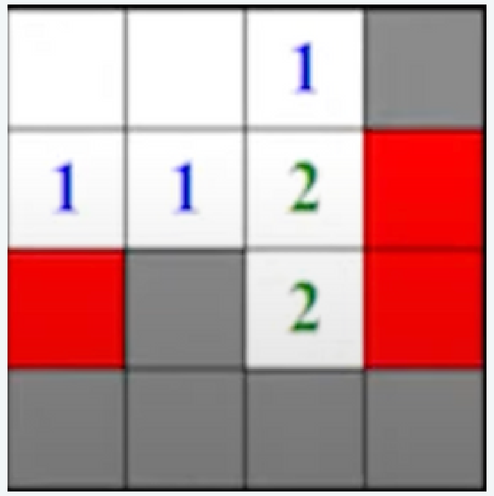
4 mines



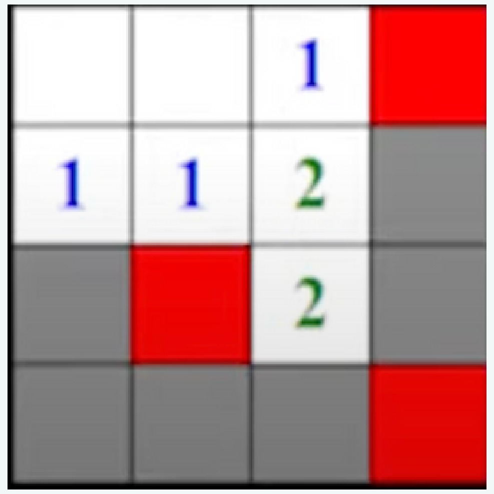
3 mines



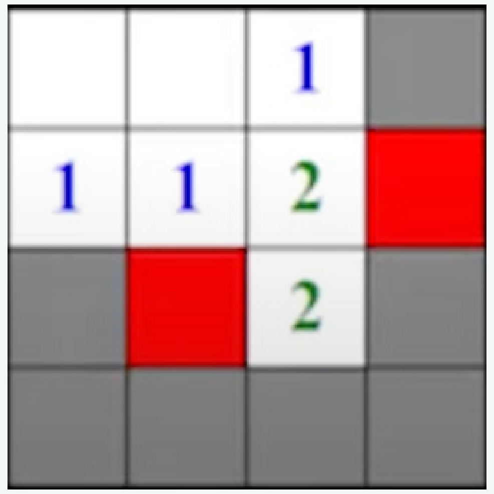
3 mines



3 mines



3 mines



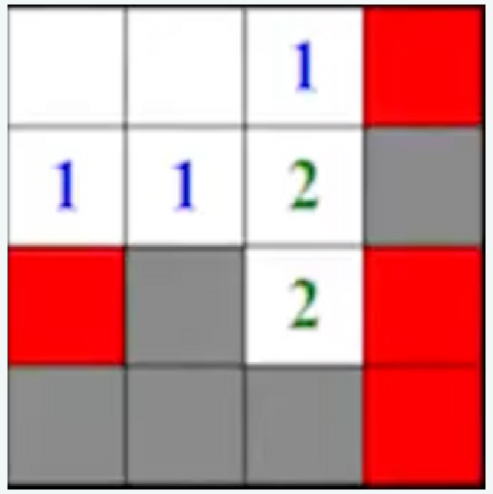
2 mines

Probabilities and Logic in Minesweeper

Conditions:

Board size: 16 x 30

Number of mines: 99



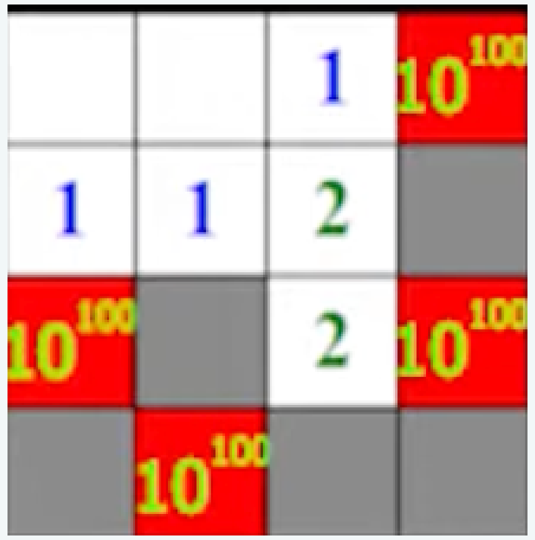
4 mines

Analytics

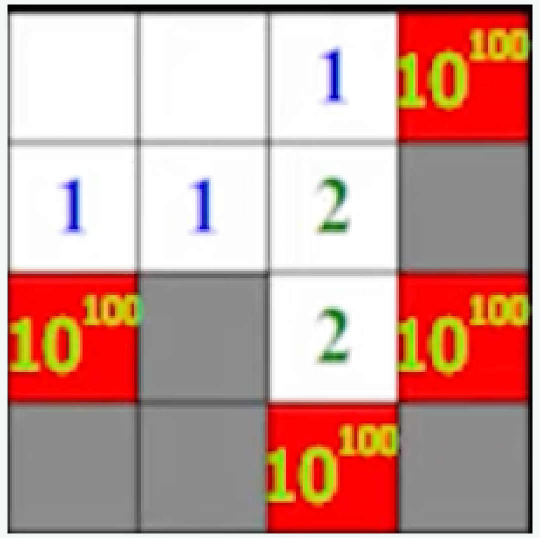
- We used here 4 mines which means the rest of the mines are spread around the board (In our case 95).
- We calculate how many arrangements there are to place the rest 95 mines.

That is: (Unbordered cell) Choose (MinesLeft)

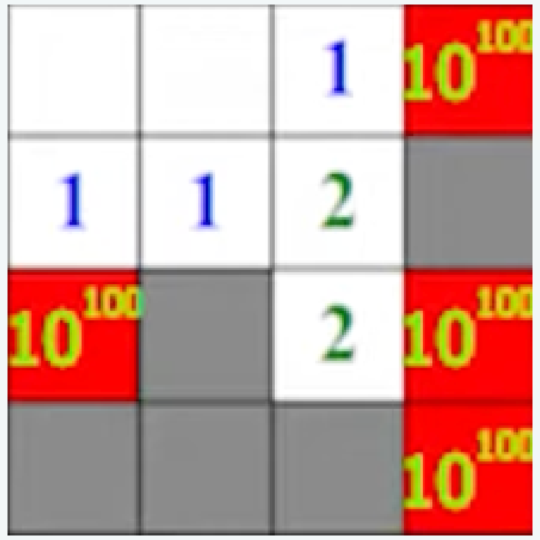
Probabilities and Logic in Minesweeper



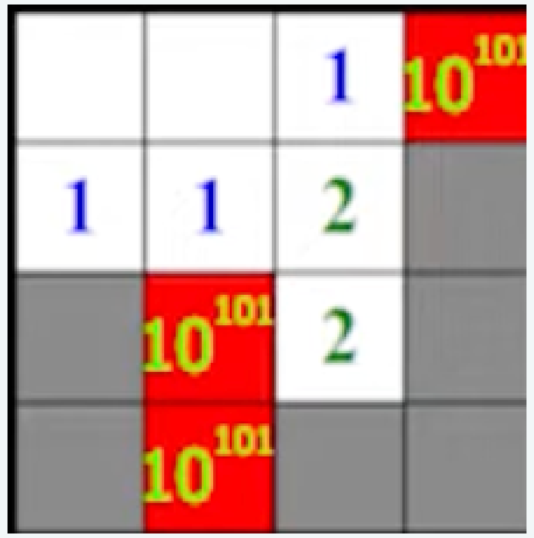
4 mines



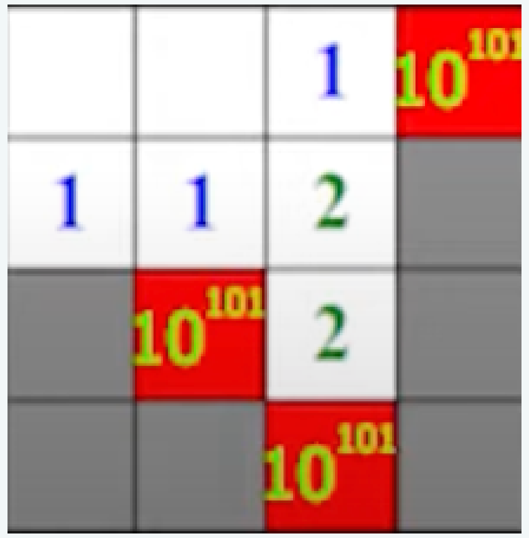
4 mines



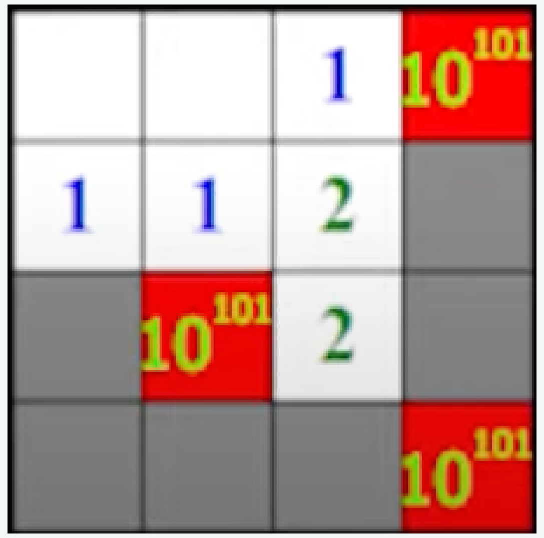
4mines



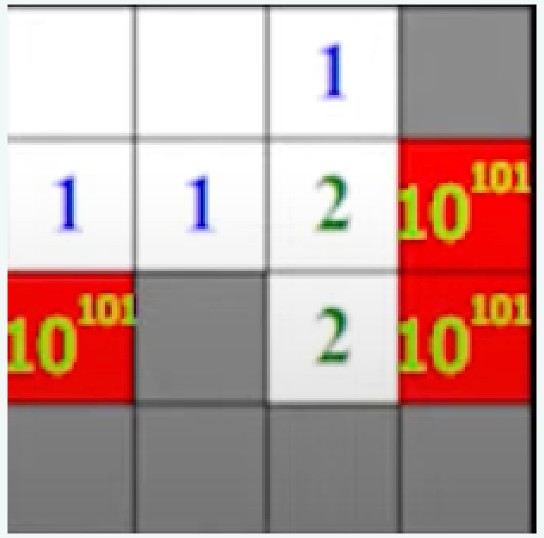
3 mines



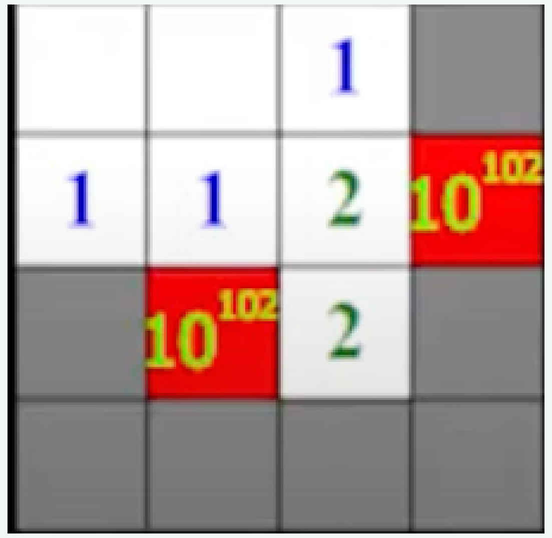
3 mines



3 mines



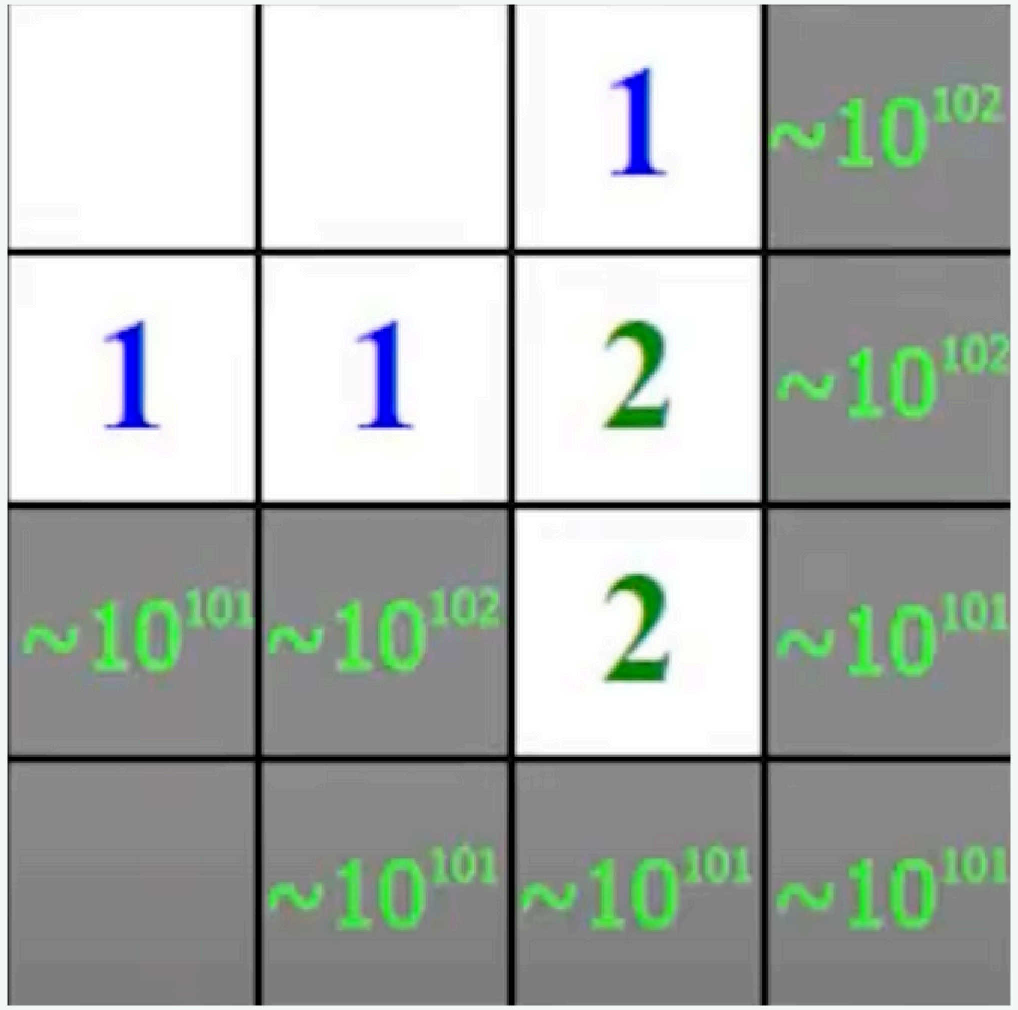
3 mines



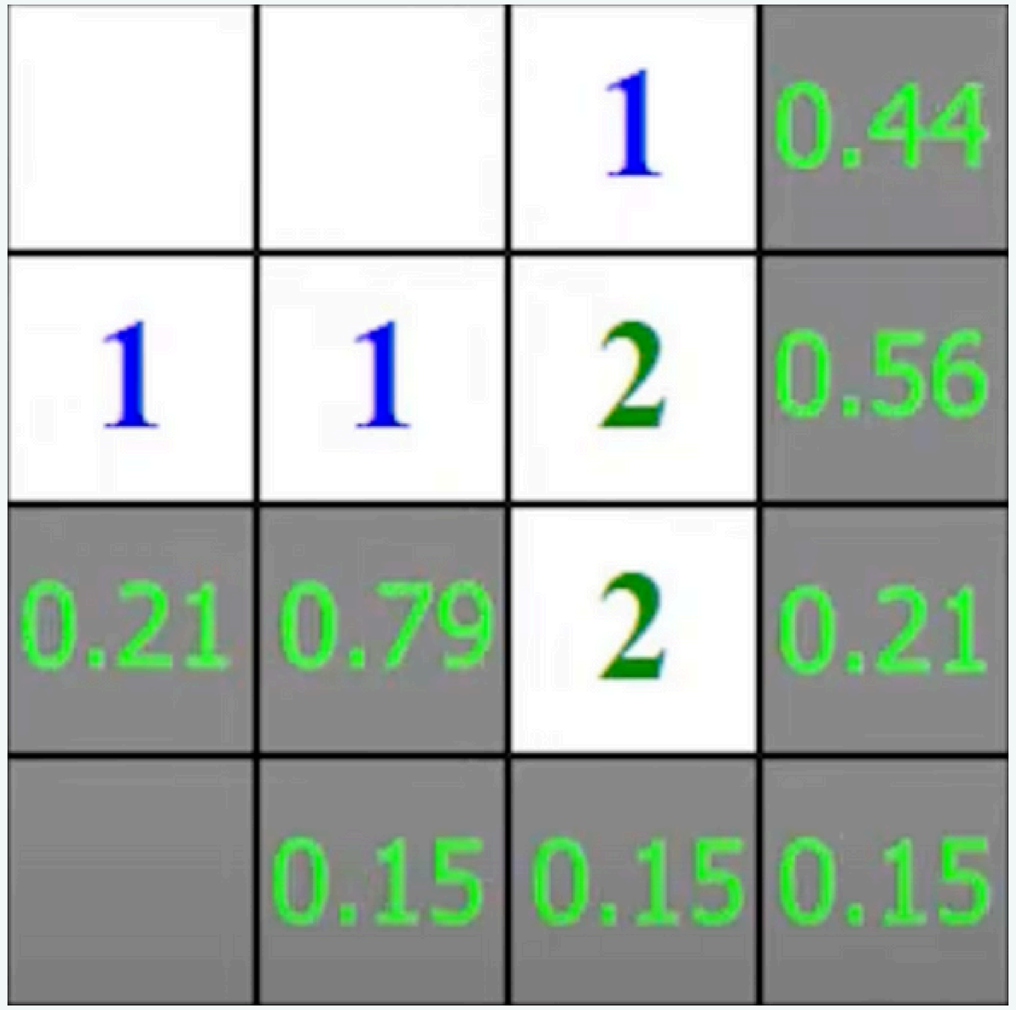
2 mines

(465) Choose (95) => 8.1 X 10^100

Probabilities and Logic in Minesweeper



Total Number of Arrangements = 2.7 × 10^102



Multiply by 100 to get cell probabilities

Probabilities

Probabilities and Logic in Minesweeper

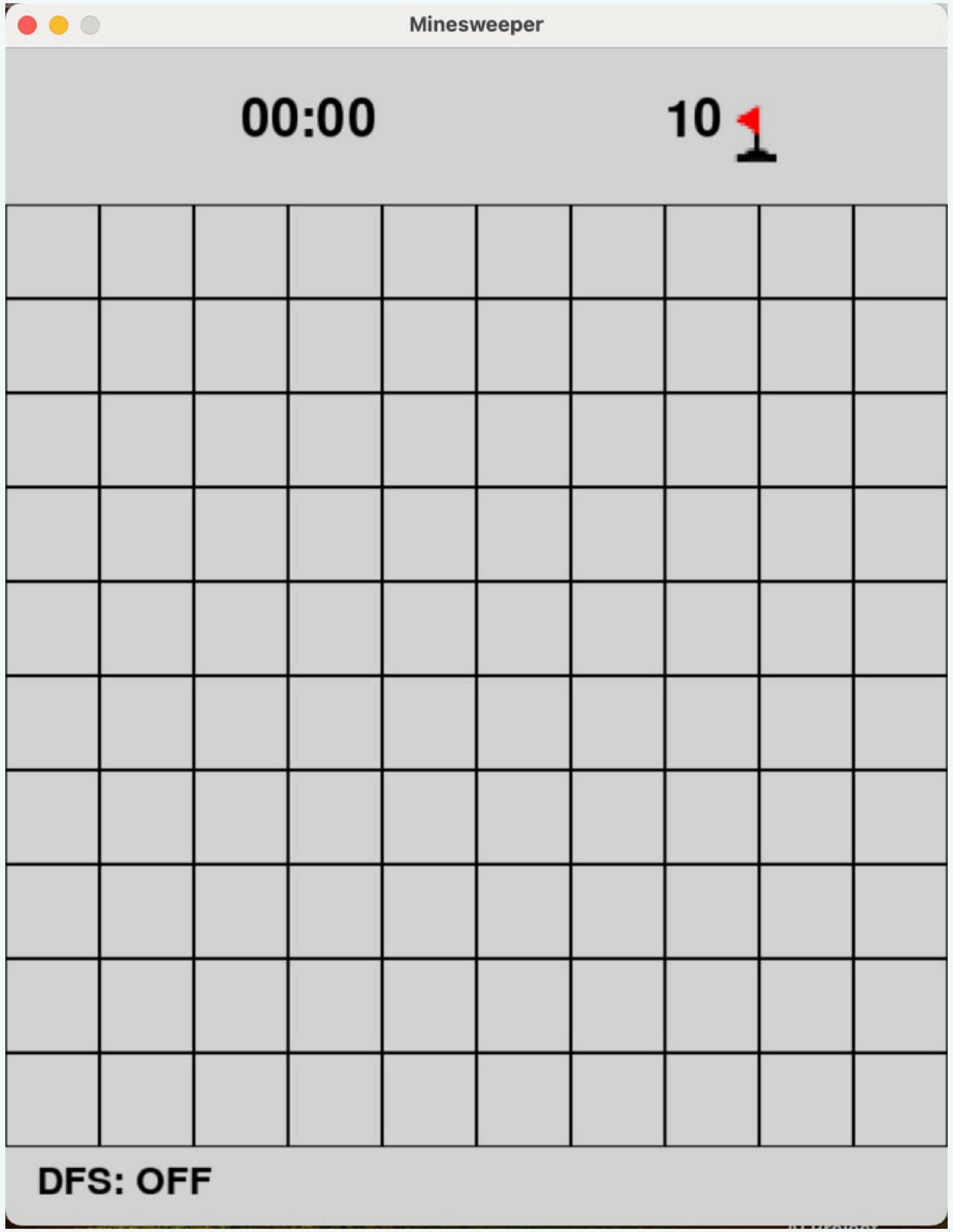
Advantages

-Handles uncertainty
-Focuses on areas of interest

Disadvantages

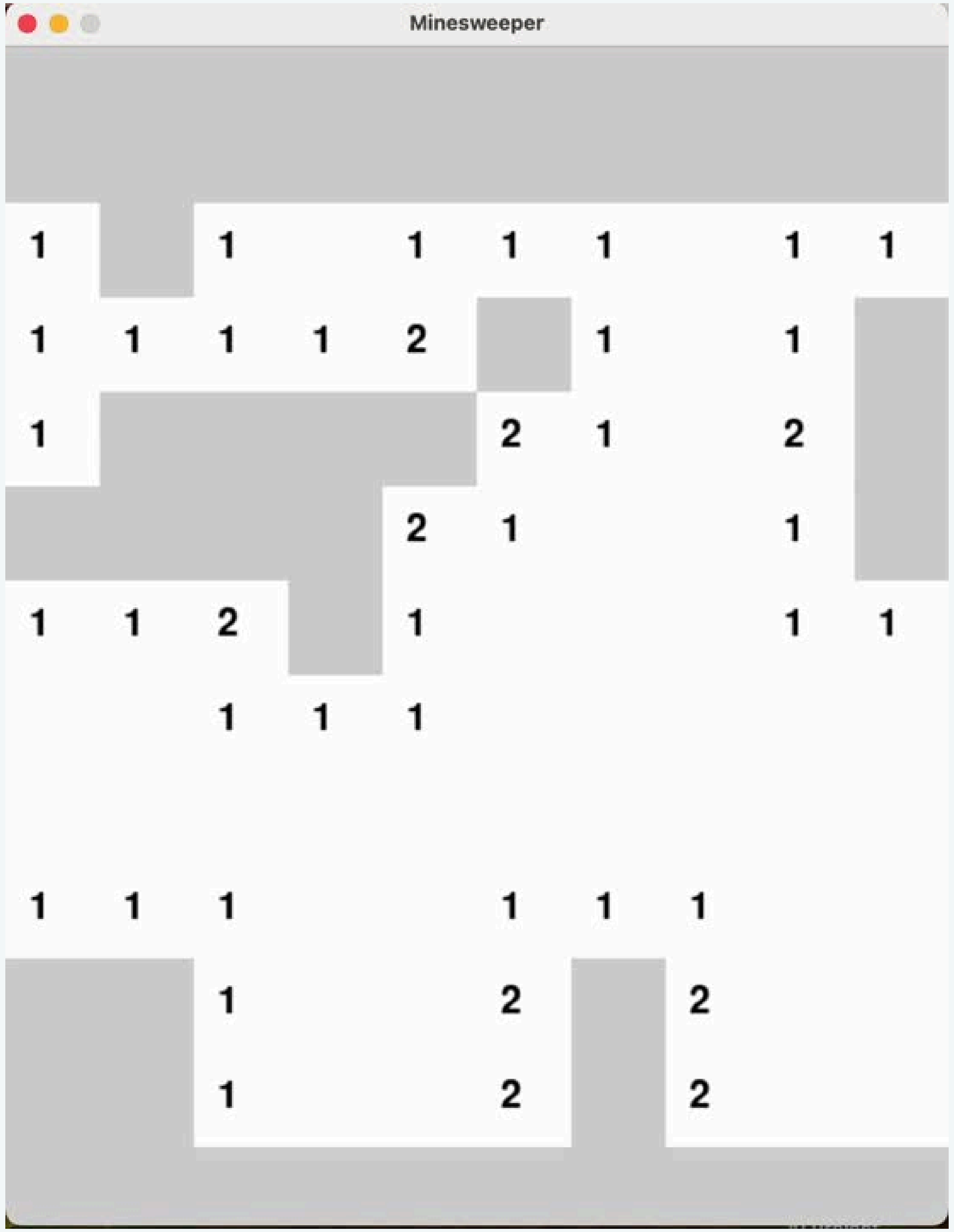
-Cannot guarantee solutions
-Complex implementation for accurate probability calculations

DFS

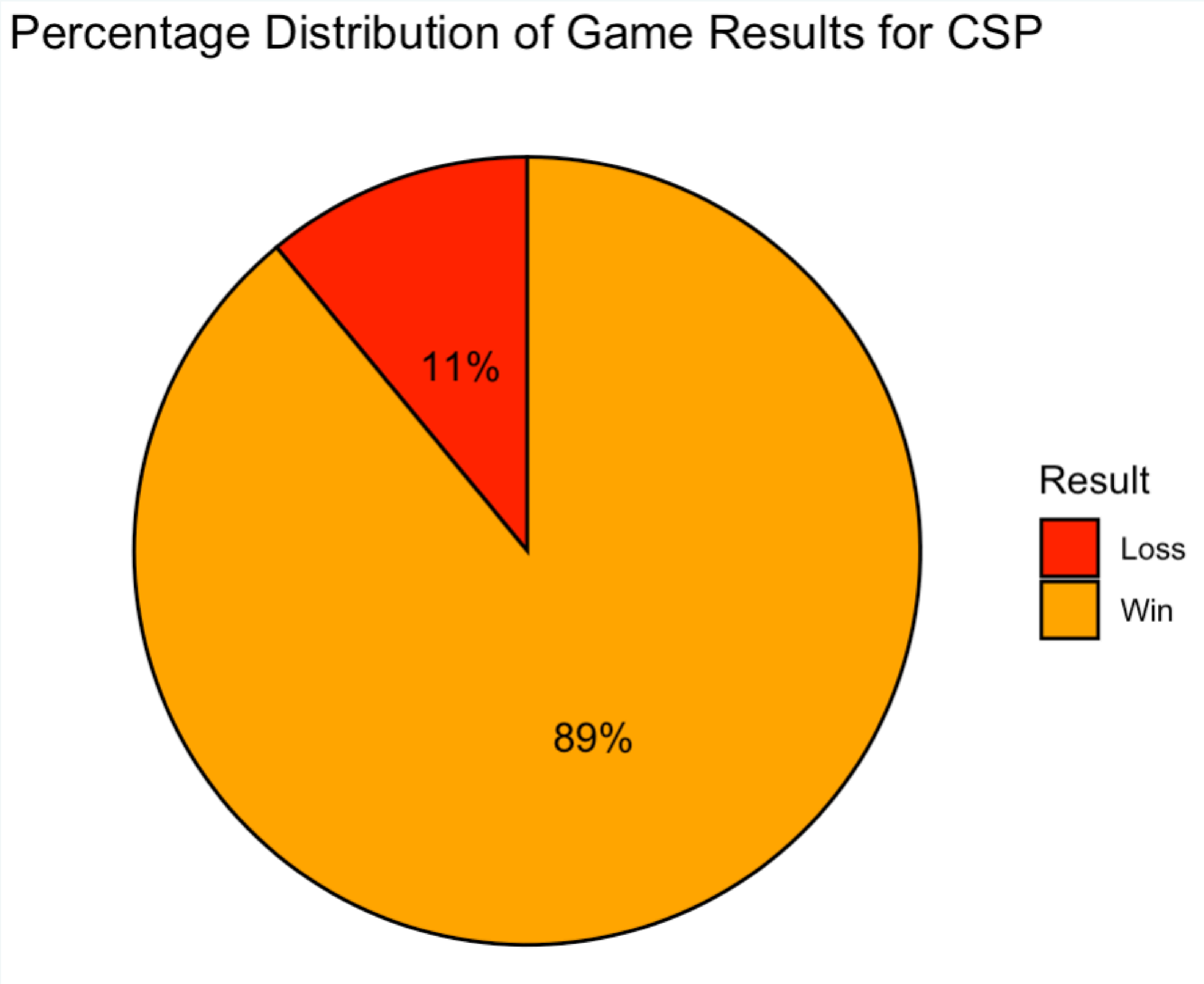


Time complexity:
O(M x N)

Space complexity:
O(M x N)



Analyses for DFS method

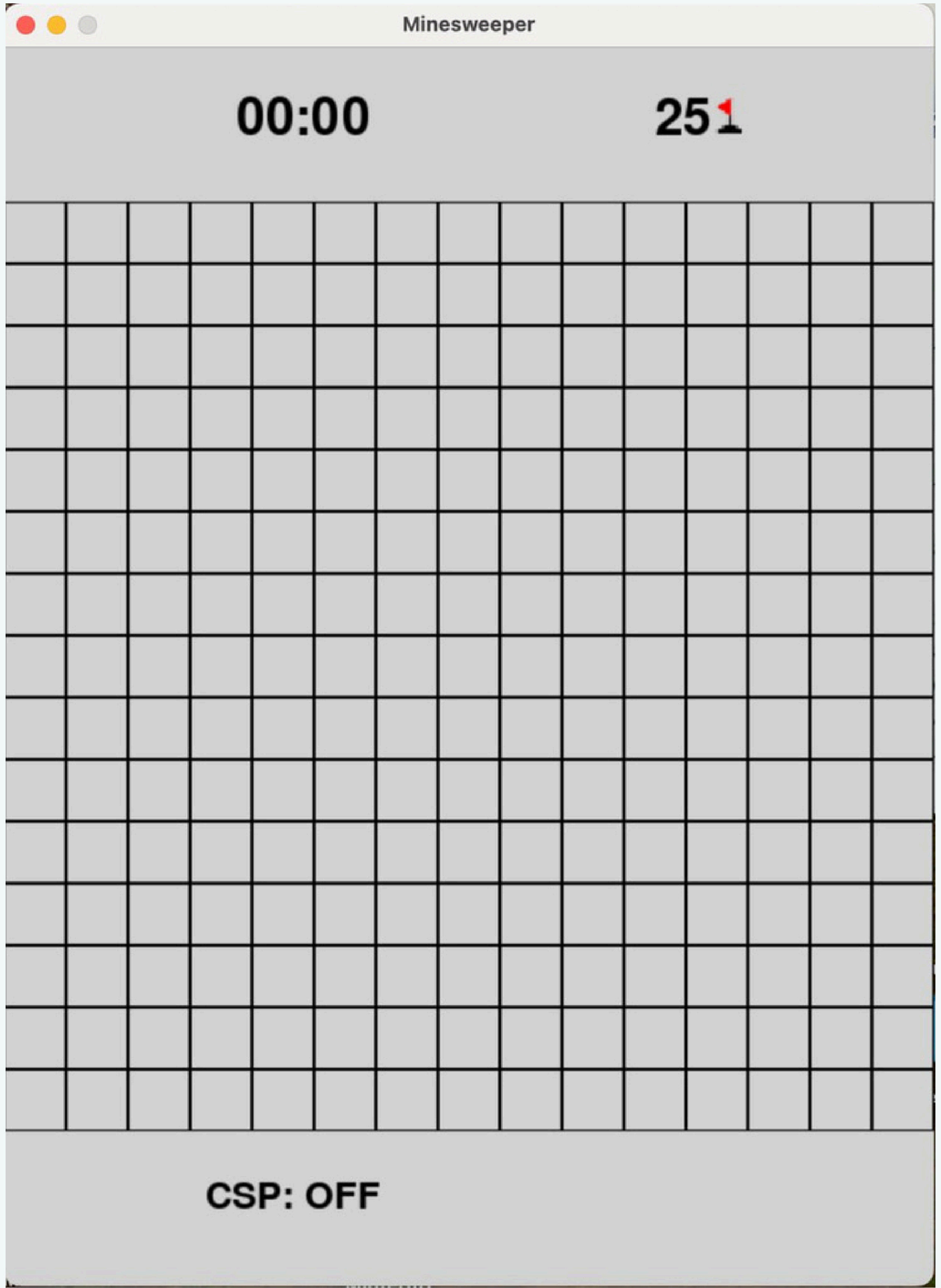


A subset of the Minesweeper DFS dataset (500 rows in total), showing Game IDs, results (Win/Lose), completion times and number od steps taken to solve the game. The analysis is based on this dataset.

GameID <dbl>	Result <chr>	Time <dbl>	Steps <dbl>
1	win	11	214
2	win	11	216
3	win	11	213
4	win	11	216
5	win	11	216
6	win	11	214
7	win	11	216
8	win	9	187
9	win	10	211
10	win	11	216

Average Winning Time: Approximately 10.78 seconds.

CSP backtracking



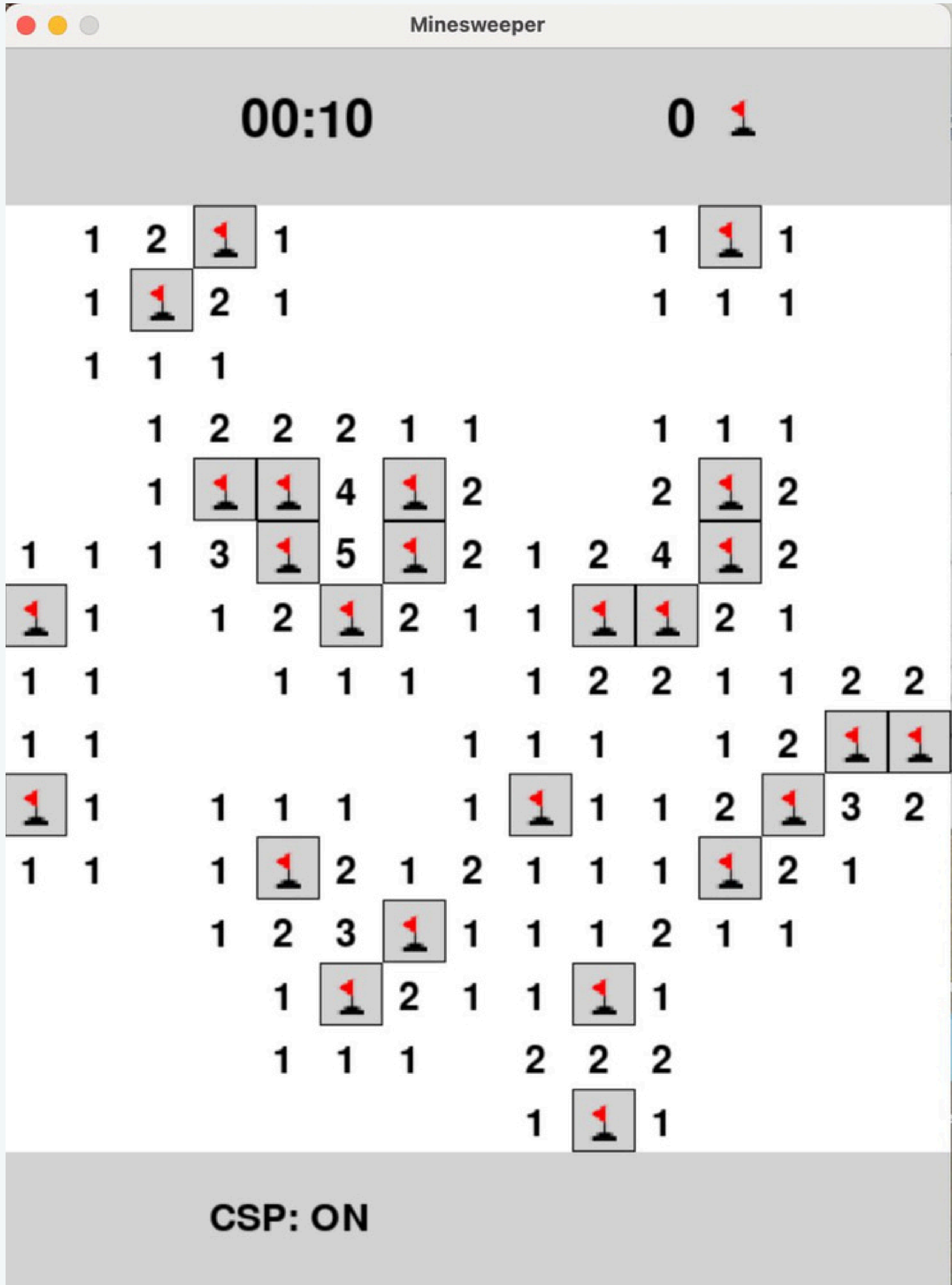
Variables:
the uncovered squares

Domain:
{0, 1}

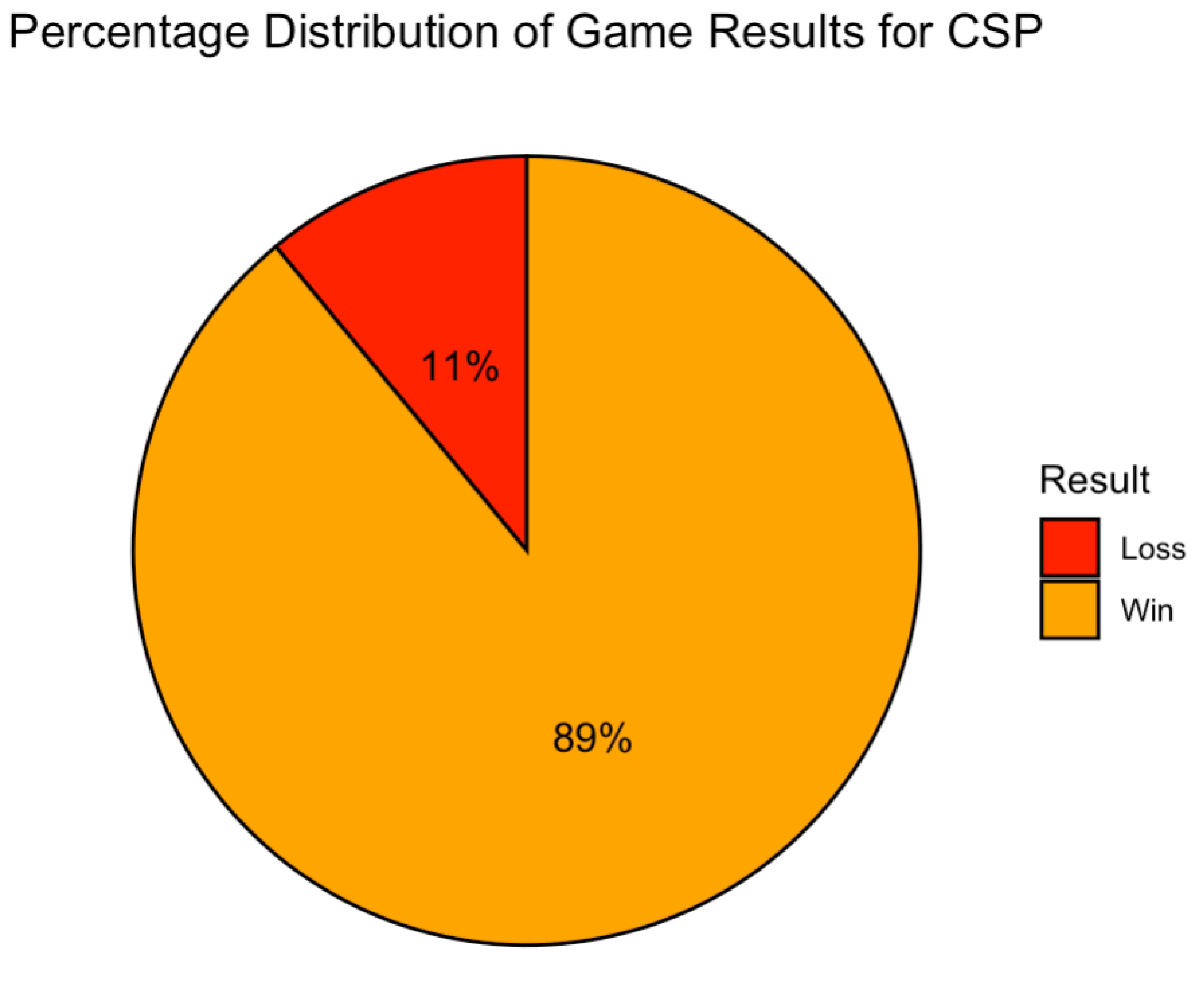
Constraint:
{0, 1, 2, 3, 4, 5, 6, 7, 8}

Time complexity:
O(2^n)

Space complexity:
O(n x d)



Analyses for CSP method



A subset of the Minesweeper DFS dataset (500 rows in total), showing Game IDs, results (Win/Lose), and completion times. The analysis is based on this dataset.

GameID <dbl>	Result <chr>	Time <dbl>
1	Win	124
2	Win	246
3	Win	207
4	Win	51
5	Win	163
6	Win	51
7	Win	135
8	Win	237
9	Win	163
10	Win	81

Average Winning Time: Approximately 142.7843 seconds.

Minesweeper to Conjunctive Normal Form Satisfiability Problem(CNF-SAT)

1. Representing Minesweeper as a Logic Problem

2. Variables and Propositions

Define a binary variable $X_{i,j}$, for each cell (i, j) on the Minesweeper grid

- *$X_{i,j}$= 1 if there is a mine in cell (i, j)*
- *$X_{i,j}$= 0 if there is no mine.*

Minesweeper to Conjunctive Normal Form Satisfiability Problem(CNF-SAT)

3. Expressing Constraints as CNF Clauses

Neighborhood Constraint

1. For each cell (i, j) with a clue c, define its neighborhood $N(i, j)$ as the set of cells adjacent to (i, j).
2. Write a constraint that says exactly c cells in $N(i, j)$ should contain mines.

To express "exactly c" in CNF, use the following approach:

1. **At least c mines**: This requires selecting c cells among $N(i, j)$ and setting each subset's mines to satisfy the "at least" condition.
2. **At most c mines**: Use combinations to exclude any configuration that has more than c mines.

4. Building CNF Clauses for Each Cell

For each cell (i, j) with a clue c, convert the "exactly c" mine constraint into CNF by combining:

1. Clauses that encode "at least c".
2. Clauses that encode "at most c".

Minesweeper to Conjunctive Normal Form Satisfiability Problem(CNF-SAT)

For a small 2×2 board:

- Suppose cell (1,1) has a clue "1," meaning exactly one of its neighbors should contain a mine.
- If cell (1,1) has three neighbors, let's label the neighboring variables $x_{1,2}, x_{2,1}, x_{2,2}$.

To encode "exactly 1 mine," use the following clauses:

1. At least 1 mine: $(x_{1,2} \vee x_{2,1} \vee x_{2,2})$.
2. At most 1 mine: $(\neg x_{1,2} \vee \neg x_{2,1}), (\neg x_{1,2} \vee \neg x_{2,2}), (\neg x_{2,1} \vee \neg x_{2,2})$.

This results in the CNF:

$$(x_{1,2} \vee x_{2,1} \vee x_{2,2}) \wedge (\neg x_{1,2} \vee \neg x_{2,1}) \wedge (\neg x_{1,2} \vee \neg x_{2,2}) \wedge (\neg x_{2,1} \vee \neg x_{2,2})$$

Minesweeper to Conjunctive Normal Form Satisfiability Problem(CNF-SAT)

Advantages

- Guarantees global consistency
- Efficient solvers available

Disadvantages

- Encoding complexity
- Scalability issues for large boards

Conclusion

- **DFS:**
  - fast for simple cases,
  - has a low memory usage,
  - is easy to implement,
  - losses if the first opened cell is a mine.
- **CSP Backtracking:**
  - handles uncertainty,
  - guarantees solutions, and logical approach,
  - slower for large grids,
  - computational complex requirements.
- **Probabilities and Logic:**
  - provides the best guess
  - won’t guarantee a correct solution,
  - don’t ensure completeness.
- **CNF-SAT:** yielding globally valid solutions.
- **Modern SAT:** solvers are optimized for handling large logical constraints.

THANK YOU!

References

- game. (2024). In Merriam-Webster Dictionary. Retrieved from https://www.merriam-webster.com/dictionary/game
- Giannone, P. (n.d.). Click, boom, and traverse: The deep world of Minesweeper's DFS. Medium. Retrieved from https://medium.com/@p.giannone333/click-boom-and-traverse-the-deep-world-of-minesweepers-dfs-e9ba42f7e7a9
- Mendelssohn, E. (2004). Introducing game theory and its applications. Taylor & Francis.
- Munir, R. (2022). Makalah IF2211 Strategi Algoritma: Analisis dan Studi Penerapan Algoritma dalam Permainan Minesweeper. Informatika STEI ITB. Retrieved from https://informatika.stei.itb.ac.id/˜rinaldi.munir/Stmik/2021-2022/Makalah/Makalah-IF2211-Stima-2022-K2%20(30).pdf
- Neumann, J., & Morgenstern, O. (1944). Theory of games and economic behavior. Princeton University Press.
- PBS. (n.d.). Nash and game theory. American Experience. Retrieved November 13, 2024, from https://www.pbs.org/wgbh/americanexperience/features/nash-game/
- Wiki, C. T. M. (n.d.). Microsoft Minesweeper. Microsoft Wiki. Retrieved from https://microsoft.fandom.com/wiki/Microsoft_Minesweeper
- The Editors of Encyclopaedia Britannica. (2024, November 15). NP-complete problem | Definition, Examples, & Facts. Encyclopedia Britannica. https://www.britannica.com/science/NP-complete-problem
- Nakov, P., & Wei, Z. (2003). Minesweeper,# Minesweeper. Unpublished Manuscript, Available at: http://www. minesweeper. info/articles/Minesweeper (Nakov, Wei). pdf.
- Fowler, A., & Young, A. (2004). Minesweeper: A statistical and computational analysis.