

# Project 2: Minesweeper

## 1 Description

You are asked to build a minesweeper solver by using CNF as described below:

Given a matrix of size  $m \times n$ , where each cell will be a non-negative integer or zero(empty cell). Each cell is considered to be adjacent to itself and 8 surrounding cells.

Minesweeper is a logic puzzle game. The game features a grid of clickable squares, with hidden "mines" scattered throughout the board. The objective is to clear the board without detonating any mines, with help from clues about the number of neighboring mines in each field.

Your puzzle needs to fill all cells with mine or not, so that the number inside each cell corresponds to the number of mines adjacent to that cell (see Figure 1)



Figure 1: Example of minesweeper puzzle

In order to solve this problem, you can consider some steps:

1. A logical variable is assigned to each cell of the matrix (If the logical variable of that cell is True, it will be mine, otherwise it will be not mine)
2. (Report) Write constraints for cells containing numbers to obtain a set of constraint clauses in CNF (note that you need to remove duplicate clauses)
3. (Implement) Generate CNFs automatically.
4. (Implement) Using the pysat library to find the value for each variable and infer the result.
5. (Implement) Apply A\* to solve the CNF.
6. (Implement) Program brute-force and backtracking algorithm to compare their speed (by measuring running time which is how long it takes for a computer to perform a specific task) and their performance with A\*.

## 2 Submitted documents

1. Source code: the entire source code with running instructions. Include folder "testcases" with 5 input test cases and 5 output.

Ex:

Input:

3, 0, 0, 0

0, 0, 0, 0

0, 0, 0, 0

Output:

3, X, 0, 0

X, X, 0, 0

0, 0, 0, 0

2. Video demo: A video recording of the process of running the tests and the results of your program.
3. Report

### 3 Requirements

No.	Criteria	Scores
1	<b>Solution description:</b> Describe the correct logical principles for generating CNFs.	30%
2	Generate CNFs automatically	10%
3	Use pysat library to solve CNFs correctly	10%
4	Implement A* to solve CNFs without a library	10%
5	Program brute-force algorithm to compare with A* (speed) Program backtracking algorithm to compare with A* (speed)	10%
6	<b>Documents and other resources that you need to write and analysis in your report:</b> Thoroughness in analysis and experimentation Give at least 5 test cases with different sizes (5x5, 9x9, 11x11, 15x15, 20x20...) to check your solution Comparing results and performance	30%

### 4 Notice

- Each group has 3 - 4 members.
- Besides the above requirements, the report must also give the following information:
  - Estimating the degree of completion level for each requirement.
  - References (if any)
- Any plagiarism, any tricks, or any lie will have a zero score for the COURSE grade