

Project 2: Gem hunter

1 Description

You are asked to build a gem hunter game by using CNF as described below:

Gem Hunter is a strategic puzzle game where players explore a grid to reveal valuable gems while avoiding dangerous traps. The objective is to reveal all the gems on the grid without triggering any traps. Players must use logic and deduction to strategically uncover tiles and mark potential trap locations. You will know the location of the trap through the tiles on the map. Each number on a tile represents the number of traps around that tile.

2	T	T	1	G
T	5	4	2	G
3	T	T	2	1
3	T	6	T	1
2	T	T	2	1

Figure 1: Example of gem hunter puzzle

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 trap, otherwise, if it is not trap, it will be gem)
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) Choose and apply an optimal algorithm 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 your chosen algorithm

2 Submitted documents

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

Symbols in file:

- T: Traps
- G: Gems (If the cell that you can determine not a trap, it is a gem)
- Number
- _ : empty cell.

Ex:

Input:

3, _, 2, _

_, _, 2, _

_, 3, 1, _

Output:

3, T, 2, G

T, T, 2, G

T, 3, 1, G

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	Solution description: 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 an optimal algorithm to solve CNFs without a library	10%
5	Program brute-force algorithm to compare with your chosen algorithm(speed) Program backtracking algorithm to compare with your chosen algorithm (speed)	10%
6	Documents and other resources that you need to write and analysis in your report: 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