

Lab: Gem hunter

1 Description

You are required to develop a gem hunter game using Conjunctive Normal Form (CNF). The game follows these rules:

- Players explore a grid to find hidden gems while avoiding traps.
- Each tile with a number represents the number of traps surrounding it. (Number from 1 - 9)
- Your task is formulating the problem as CNF constraints and solving it using logic.

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. Assign a logical variable for each cell in the grid: True (T): The cell contains a trap. False (G): The cell contains a 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) 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 using the library

2 Submitted documents

1. Source code: The entire source code with running instructions. Include a folder named "testcases" containing 3 input test cases and their corresponding 3 output files. **Note:** The generated input test cases must be solvable.

Symbols in the file:

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

Ex:

Input: The input file format must be named "input_x.txt" (with x is number of input). Ex "input_1.txt".

3, _, 2, _

_, _, 2, _

_, 3, 1, _

Output: The output file format must be named "output_x.txt" (with x is number of output). Ex "output_1.txt".

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.	20%
2	Generate CNFs automatically	10%
3	Use pysat library to solve CNFs correctly	10%
4	Program brute-force algorithm to compare with using library(speed)	10%
5	Program backtracking algorithm to compare with using library (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 3 test cases with different sizes (5x5, 11x11, 20x20) to check your solution Comparing results and performance	40%

4 Notice

- This is an individual exercise.
- Besides the above requirements, the report must also give the following information:
 - Estimating the degree of completion level for each requirement.
 - References (if any)
- AI tools are not restricted. However, students should use them wisely. Lab instructors can conduct additional oral interviews to check your work in this lab.
- Any act of plagiarism or academic dishonesty—including but not limited to copying from external sources without proper citation, fabricating data, submitting work that is not one's own, or submitting work with more than 70% similarity to someone else's—will receive 0 points for the course grade.