



**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN,
ĐHQG-TPHCM
KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN CƠ SỞ TRÍ TUỆ NHÂN TẠO
ĐỒ ÁN 2 – MINESWEEPER**

Sinh viên thực hiện:

Hà Minh Khang – 21127623

Kuo Yung Sheng – 21127684

Nguyễn Đình Vinh – 21127724

Giảng viên hướng dẫn: Nguyễn Trần Duy Minh

MỤC LỤC

1.	Mô tả	3
1.1.	Yêu cầu.....	3
1.2.	Tổng quan về chương trình	3
1.3.	Đầu vào và đầu ra	3
2.	Nguyên tắc logic để khởi tạo các mệnh đề CNF.	4
3.	Ý tưởng thuật toán	5
3.1.	Ý tưởng khởi tạo CNFs.....	5
3.2.	Ý tưởng thuật toán resolution (thay cho A^*).....	6
3.3.	Ý tưởng thuật toán dpll (backtracking)	6
3.4.	Ý tưởng thuật toán brute force	6
3.5.	Pysat solver	6
4.	Chương trình.....	7
5.	Các bộ test mẫu và kết quả	7
6.	Phân công công việc	12
7.	Đánh giá mức độ hoàn thành	12
8.	TÀI LIỆU THAM KHẢO	13

1. Mô tả

1.1. Yêu cầu

Giải quyết bài toán logic Minesweeper (dò mìn) bằng cách khởi tạo chính xác các mệnh đề CNFs. Những ô có giá trị số (0, 1, 2, ..) thể hiện có bao nhiêu mìn xung quanh (liền kề) ô đó. Trường hợp số 0 là không có mìn xung quanh ô đó. Ô chứa giá trị (-) là những ô chưa biết.

Những ô chứa giá trị số sẽ được khởi tạo các mệnh đề CNFs. Từ những mệnh đề CNFs đã khởi tạo, áp dụng thuật toán hoặc pysat để suy luận kết quả cho ô đang được xét. Nếu ô nào có mìn thì sẽ được gán giá trị True, và ngược lại không có mìn thì gán giá trị False.

1.2. Tổng quan về chương trình

Môi trường lập trình: ngôn ngữ python.

Các thư viện sử dụng: (không sử dụng các thư viện bên ngoài)

- Pysat.solver (Glucose4) để giải quyết yêu cầu bài toán.
- Time, psutil: để tính toán thời gian và bộ nhớ khi thực thi thuật toán.
- Copy: hỗ trợ cho thuật toán.

1.3. Đầu vào và đầu ra

Đầu vào (input) là một file txt bao gồm một mảng 2 chiều kích thước $n \times n$. Phần tử mảng bao gồm các giá trị số (0, 1, 2,...) hoặc (-) đã được giải thích ở trên, các phần tử được ngăn cách nhau bởi dấu “,”.

VD:

3,-,-

-,,-

-,,-

Đầu ra (output) là một file txt chứa kết quả xử lý của file đầu vào. Bao gồm các giá trị số (0, 1, 2,...), những ô không có mìn (-) và những ô là mìn (X).

VD: Sau khi áp dụng các thuật toán để giải mệnh đề CNFs cho file input trên sẽ có kết quả ghi vào file output.txt:

3,X,-

X,X,-

-, -, -

2. Nguyên tắc logic để khởi tạo các mệnh đề CNF.

Duyệt theo thứ tự từ trái sang phải và từ trên xuống dưới của mảng 2 chiều.

Vị trí của ô đang xét được chuyển đổi như sau: với mảng 2 chiều có kích thước 3x3 thì ô (0,0), (0,1), (0,2), (1,0), (1,1)... lần lượt được chuyển đổi là 1, 2, 3, 4, 5...

Nếu ô đang được xét có giá trị là số (0, 1, 2...) thì sẽ có constraint là âm tại ô đó (vì ô là số thì không có min được). VD: xét ô (1,1) có giá trị là số trong mảng 2 chiều 3x3 thì sẽ có constraint là [-5].

Sau đó duyệt các ô xung quanh (liền kề) ô đang xét.

Giả sử trường hợp sau

-	3	-
-	-	-

Ở thông tin ô (0,1) (còn gọi là ô 2) đang có giá trị là 3 và xung quanh là những ô chưa biết, ta suy luận được rằng:

Chính xác 3 trong 5 ô (1,3,4,5,6) chứa min

Điều này tương đương với phép giao của 2 mệnh đề sau:

- A: Có ít nhất 3 trong 5 ô chứa min
- B: Có nhiều nhất 3 trong 5 ô chứa min

Với A, ta chọn ngẫu nhiên 3 ô trong 5 ô, thì chắc chắn trong 3 được chọn có ít nhất 1 ô là min. Vậy có $5C3 = 10$ trường hợp.

[3, 6, 5], [3, 6, 4], [3, 6, 1], [3, 5, 4], [3, 5, 1], [3, 4, 1], [6, 5, 4], [6, 5, 1], [6, 4, 1],
[5, 4, 1]

Ví dụ cho 2 trường hợp đầu tiên

$[3, 6, 5], [3, 6, 4]$ có nghĩa là $(3 \vee 6 \vee 5) \wedge (3 \vee 6 \vee 4)$. Giá trị của các ô là dương biểu thị cho việc có mìn.

Với B, ta sẽ chuyển đổi thành

- B': Có ít nhất 2 trong 5 ô không chứa mìn

Chọn ngẫu nhiên 4 ô trong 5 ô, thì trong 4 ô được chọn có ít nhất 1 ô không có mìn. Vậy có $5C4 = 5$ trường hợp

$[-3, -6, -5, -4], [-3, -6, -5, -1], [-3, -6, -4, -1], [-3, -5, -4, -1], [-6, -5, -4, -1]$

Tương tự như A, nhưng giá trị của ô là âm để biểu thị cho việc ô đó không có mìn.

Sau khi có đầy đủ các mệnh đề có sẵn, sử dụng thuật toán để giao các mệnh đề với nhau tạo nên kết quả cuối cùng.

3. Ý tưởng thuật toán

3.1. Ý tưởng khởi tạo CNFs

Ô đang xét là giá trị số thì ô đó là False

Như đã trình bày ở trên, một ô có giá trị là k có n ô xung quanh thì

Có chính xác k mìn trong n ô chứa mìn

Tương đương với phép giao 2 mệnh đề sau

A: Có ít nhất k trong n ô xung quanh chứa mìn

B: Có nhiều nhất k trong n ô xung quanh chứa mìn

Ta phải chuyển đổi về dạng

A': Chọn ngẫu nhiên k ô trong n ô, thì trong k ô đó có ít nhất 1 ô là mìn.

B': Chọn ngẫu nhiên k+1 ô trong n ô, thì trong (k+1) ô đó có ít nhất 1 ô không phải là mìn.

Ta có suy luận để ra các mệnh đề bằng cách đệ quy dựa trên số lượng tổ hợp khi chọn ngẫu nhiên. Mỗi mệnh đề là mỗi kết quả tổ hợp khác nhau.

Nếu kết quả của tổ hợp là 1, tức là chọn k ô trong n ô ($k = n$) thì có thể suy luận được ngay là trong k ô đó tất cả đều có bom. Ví dụ với ma trận 2×2 , ô 0 có giá trị là 3 là ô 1,2,3 là (-) thì suy luận được ô 1, 2, 3 đều có bom.

3.2. Ý tưởng thuật toán resolution (thay cho A*)

def apply_resolution(cnf_clauses): Hàm Resolution.

1. Chạy vòng lặp đến khi còn 1 clauses hoặc đến khi không còn clause nào được tính và các clauses vẫn như cũ (hội tụ).
2. Giải các clauses đơn biến trước(hàm unit_propagation). Lấy một clause đơn biến A(-A là không có boom, A là có boom), nếu clause nào chứa mệnh đề A (mệnh đề đúng) thì xóa clause đó. Còn nếu chứa -A(mệnh đề sai) thì xóa -A khỏi clause. Lặp lại đến khi không còn clause đơn biến.
3. Resolve các clauses có số phần tử là 3 với các clauses có số phần tử là 2 bị duplicated. Nếu kết quả của resolution là clause có số phần tử là 1. Thì phần tử đó dc thêm vào tập clause, vì phần tử đó là sai nên ta sẽ lấy trái dấu > 0 thì gán là False, < 0 thì gán là True. Và xử lý tương tự như bước 2.
VD $[2,3,4],[3,4] \Rightarrow [2]$
4. Trả về assignment (kết quả).

3.3. Ý tưởng thuật toán dp11 (backtracking)

1. Đầu tiên, thuật toán sẽ lấy các giá trị đã được gán (các mệnh đề chỉ có độ dài bằng 1) (VD: $[-1]$, $[2]$, ..). Nếu các mệnh đề không có xung đột (VD: không có $[2]$ $[-2]$), thì sẽ được gán vào kết quả.
2. Gán ngẫu nhiên giá trị đầu tiên tại clause đầu tiên là True(gọi giá trị đó là A). Thêm clause mới(A) vào. Gọi đến hàm dp11 để backtrack để giải tiếp clauses mới, nếu không có kết quả thì giá trị A: True là sai, gán giá trị A: False, cập nhật lại assignment.
3. Gán ngẫu nhiên giá trị đầu tiên tại clause đầu tiên là False(gọi giá trị đó là A). Thêm clause mới(A) vào. Gọi đến hàm dp11 để backtrack để giải tiếp clauses mới, nếu không có kết quả thì giá trị A: False là sai, gán giá trị A: True, cập nhật lại assignment.
4. Nếu tất cả các biến đã được gán thì trả về kết quả, ngược lại trả về rỗng.

3.4. Ý tưởng thuật toán brute force

1. Brute force sẽ gọi đệ quy tất cả phép gán có thể từ 1 tới n. Hàm sẽ lần lượt gán giá trị False và True và gọi đệ quy cho mệnh đề tiếp theo. Nếu phép gán cho mệnh đề thỏa mãn tất cả mệnh đề còn lại thì trả về True, ngược lại là False
2. Sau khi duyệt xong, hàm sẽ trả về danh sách sau khi gán.

3.5. Pysat solver

Glucose4, gọi solver.add_clause để thêm các mệnh đề vào solver, sau đó gọi hàm get_model để pysat đưa ra kết quả.

Thư viện pysat.solver sẽ có thêm bước dò ngẫu nhiên những ô chưa xác định được, điều mà các thuật toán trên chưa thể làm được. Cụ thể hơn được nêu ở phần sau.

4. Chương trình

Chương trình bao gồm hàm main và tất cả các thuật toán. Chương trình sẽ yêu cầu người dùng nhập vào tên file, và lựa chọn (1. Resolution, 2. Brute force, 3. Backtracking, 4. Pysat).

Đối với file nhập vào, bao gồm 5 file text có định dạng (input_size.txt) với size là kích thước của mảng. VD: mảng 15x15 có file là input_15.txt

Với file output cũng là file text có định dạng như file input. (VD: output_15.txt).

5. Các bộ test mẫu và kết quả

Video demo: <https://youtu.be/TB27eR8eQf0>

Lưu ý: Các khung màu đỏ để chỉ rõ ra sự khác nhau giữa pysat và các thuật toán khác. Ở trường hợp 9x9 trở lên, brute force không có đáp án do thời gian chạy vượt qua ngưỡng cho phép.

Test case 1: ma trận 5x5

	Output	
<div><div>-,-,1,2,1</div><div>-,-,-,-,-</div><div>1,2,-,-,2</div><div>0,1,2,-,1</div><div>0,1,-,-,1</div></div> <div>Input</div>	<div><div>-,-,1,2,1</div><div>X,-,X,-,X</div><div>1,2,-,-,2</div><div>0,1,2,X,1</div><div>0,1,X,-,1</div></div> <div>Pysat</div>	<div><div>-,-,1,2,1</div><div>-,-,-,-,-</div><div>1,2,-,-,2</div><div>0,1,2,-,1</div><div>0,1,-,-,1</div></div> <div>Các thuật toán còn lại (chung output)</div>

Test case 2: ma trận 9x9

	Output
--	--------

<pre> -, -, 1, -, -, 2, -, 1, 0 -, -, -, -, -, -, -, 1, 0 -, -, -, 2, -, 2, 1, 1, 0 1, -, 1, -, 2, -, -, 3, 1 -, 2, 2, -, -, -, -, 3, - 1, 2, -, 1, 1, 2, 2, 2, - 0, 1, 1, 1, 0, 0, 0, 1, 1 2, 2, 1, 0, 1, 1, 1, 2, - -, -, 1, 0, 1, -, 1, -, - </pre>	<pre> -, X, 1, -, -, 2, X, 1, 0 -, -, -, -, X, -, -, 1, 0 -, -, -, 2, -, 2, 1, 1, 0 1, -, 1, X, 2, -, X, 3, 1 X, 2, 2, -, -, X, X, 3, X 1, 2, X, 1, 1, 2, 2, 2, - 0, 1, 1, 1, 0, 0, 0, 1, 1 2, 2, 1, 0, 1, 1, 1, 2, X X, X, 1, 0, 1, X, 1, -, X </pre>	<pre> -, -, 1, -, -, 2, X, 1, 0 -, -, -, -, X, -, -, 1, 0 -, -, -, 2, -, 2, 1, 1, 0 1, -, 1, X, 2, -, X, 3, 1 X, 2, 2, -, -, X, X, 3, X 1, 2, X, 1, 1, 2, 2, 2, - 0, 1, 1, 1, 0, 0, 0, 1, 1 2, 2, 1, 0, 1, 1, 1, 2, X X, X, 1, 0, 1, X, 1, -, X </pre>
Input	Pysat	Resolution và backtracking

Test case 3: ma trận 11x11

Input	Output	
<pre> -, -, 2, -, 3, -, 2, -, 1, 0, 0 2, -, -, -, -, 2, 1, 2, 1, 1 -, 2, 2, 2, 2, 1, 0, 1, 2, - -, 1, 0, 0, 0, 0, 0, 2, -, - 1, 1, 1, 0, 1, 1, 1, 2, -, 3, - 1, -, 2, 2, 3, -, -, -, 3, 3, 1 1, 2, -, -, -, -, 4, -, -, 1, 0 -, 1, 1, 2, 2, 2, -, -, 4, 3, 2 1, 1, 0, 1, 1, 1, 1, 2, 3, - -, 1, 0, 1, -, 1, 0, 1, -, -, - 1, 1, 0, 1, 1, 1, 0, 1, 1, 2, - </pre>	<pre> -, -, 2, -, 3, X, 2, X, 1, 0, 0 2, X, -, X, X, 2, 2, 1, 2, 1, 1 X, 2, 2, 2, 2, 1, 0, 1, 2, X, - -, 1, 0, 0, 0, 0, 0, 2, X, -, - 1, 1, 1, 0, 1, 1, 1, 2, X, 3, X 1, X, 2, 2, 3, X, -, -, 3, 3, 1 1, 2, X, -, X, X, 4, X, X, 1, 0 -, 1, 1, 2, 2, 2, -, X, 4, 3, 2 1, 1, 0, 1, 1, 1, 1, 2, 3, X, X X, 1, 0, 1, X, 1, 0, 1, X, -, X 1, 1, 0, 1, 1, 1, 0, 1, 1, 2, - </pre>	<pre> -, -, 2, -, 3, X, 2, X, 1, 0, 0 2, X, -, X, X, 2, 2, 1, 2, 1, 1 X, 2, 2, 2, 2, 1, 0, 1, 2, X, - -, 1, 0, 0, 0, 0, 0, 2, X, -, - 1, 1, 1, 0, 1, 1, 1, 2, X, 3, X 1, X, 2, 2, 3, X, -, -, 3, 3, 1 1, 2, X, -, X, X, 4, X, X, 1, 0 -, 1, 1, 2, 2, 2, -, X, 4, 3, 2 1, 1, 0, 1, 1, 1, 1, 2, 3, X, X X, 1, 0, 1, X, 1, 0, 1, X, -, - 1, 1, 0, 1, 1, 1, 0, 1, 1, 2, - </pre>
	Pysat	Resolution và backtracking

Test case 4: ma trận 15x15

	Output	
<pre> 0, 0, 0, 0, 1, 1, 2, -, -, 2, -, -, 1, 0, 0 1, 1, 0, 0, 1, -, -, -, -, 3, -, -, 2, 1, 0 -, 2, 1, 1, 1, 1, 3, 2, -, -, -, 2, -, 1, 0 -, 3, -, -, 2, 3, 4, -, 4, 3, 2, 2, 2, 2, 1 1, -, -, -, -, -, -, -, 2, -, 1, 1, -, 1 1, 1, 1, 2, -, -, 5, -, 3, 2, 1, -, 2, 2, 1 0, 1, 1, 2, 1, 2, -, 3, 2, -, 1, 2, -, 2, - 0, 1, -, 1, 0, 2, -, -, 2, -, -, -, 3, -, - 0, 2, -, 2, 0, 1, -, -, 2, 1, 2, 3, -, 2, - 0, 1, -, 1, 0, 2, -, -, 2, -, 1, -, -, 2, 1 0, 1, 1, 1, 1, 2, -, -, 1, 1, 2, 2, -, -, - 0, 0, 1, -, 3, -, 4, -, 1, -, -, 2, -, 2, 1 1, 1, 2, -, -, 3, -, 2, -, 1, 1, 2, -, 2, - 1, -, 3, 3, 3, 2, 2, -, 2, 1, 0, 1, 2, -, 1 -, 1, 2, -, 1, 0, 1, 2, -, 1, 0, 0, 1, -, - </pre>	<pre> 0, 0, 0, 0, 1, 1, 2, -, X, 2, -, X, 1, 0, 0 1, 1, 0, 0, 1, X, -, X, -, 3, X, -, 2, 1, 0 X, 2, 1, 1, 1, 1, 3, 2, -, X, -, 2, X, 1, 0 -, 3, X, -, 2, 3, 4, X, 4, 3, 2, 2, 2, 2, 1 1, X, -, -, X, X, X, X, X, 2, X, 1, 1, X, 1 1, 1, 1, 2, X, -, 5, X, 3, 2, 1, -, 2, 2, 1 0, 1, 1, 2, 1, 2, X, 3, 2, -, 1, 2, X, 2, - 0, 1, X, 1, 0, 2, -, X, 2, -, X, -, 3, X, - 0, 2, -, 2, 0, 1, X, X, 2, 1, 2, 3, X, 2, - 0, 1, X, 1, 0, 2, -, -, 2, -, 1, X, -, 2, 1 0, 1, 1, 1, 1, 2, X, X, 1, 1, 2, 2, -, X, - 0, 0, 1, -, 3, X, 4, -, 1, -, X, 2, -, 2, 1 1, 1, 2, X, X, 3, X, 2, -, 1, 1, 2, X, 2, - 1, X, 3, 3, 3, 2, 2, X, 2, 1, 0, 1, 2, X, 1 -, 1, 2, X, 1, 0, 1, 2, X, 1, 0, 0, 1, -, - </pre>	<pre> 0, 0, 0, 0, 1, 1, 2, -, -, 2, -, -, 1, 0, 0 1, 1, 0, 0, 1, X, -, X, -, 3, -, -, 2, 1, 0 X, 2, 1, 1, 1, 1, 3, 2, -, X, -, 2, X, 1, 0 -, 3, X, -, 2, 3, 4, X, 4, 3, 2, 2, 2, 2, 1 1, X, -, -, X, X, X, X, X, 2, X, 1, 1, X, 1 1, 1, 1, 2, X, -, 5, X, 3, 2, 1, -, 2, 2, 1 0, 1, 1, 2, 1, 2, X, 3, 2, -, 1, 2, X, 2, - 0, 1, X, 1, 0, 2, -, X, 2, -, X, -, 3, X, - 0, 2, -, 2, 0, 1, X, X, 2, 1, 2, 3, X, 2, - 0, 1, X, 1, 0, 2, -, -, 2, -, 1, X, -, 2, 1 0, 1, 1, 1, 1, 2, X, X, 1, 1, 2, 2, -, -, - 0, 0, 1, -, 3, X, 4, -, 1, -, X, 2, -, 2, 1 1, 1, 2, X, X, 3, X, 2, -, 1, 1, 2, X, 2, X 1, X, 3, 3, 3, 2, 2, X, 2, 1, 0, 1, 2, -, 1 -, 1, 2, X, 1, 0, 1, 2, X, 1, 0, 0, 1, -, - </pre>
Input	Pysat	Resolution

0,0,0,0,1,1,2,-,-,2,-,-,1,0,0
1,1,0,0,1,X,-,X,-,3,-,-,2,1,0
X,2,1,1,1,1,3,2,-,X,-,2,X,1,0
-,-,3,X,-,2,3,4,X,4,3,2,2,2,1
1,X,-,-,X,X,X,X,2,X,1,1,X,1
1,1,1,2,X,-,5,X,3,2,1,-,2,2,1
0,1,1,2,1,2,X,3,2,-,1,2,X,2,-
0,1,X,1,0,2,-,X,2,-,X,-,3,X,-
0,2,-,2,0,1,X,X,2,1,2,3,X,2,-
0,1,X,1,0,2,-,-,2,-,1,X,-,2,1
0,1,1,1,1,2,X,X,1,1,2,2,-,-,-
0,0,1,-,3,X,4,-,1,-,X,2,-,2,1
1,1,2,X,X,3,X,2,-,1,1,2,X,2,-
1,X,3,3,3,2,2,X,2,1,0,1,2,-,1
-,-,1,2,X,1,0,1,2,X,1,0,0,1,-,-

Backtracking

Test case 5: ma trận 20x20

	Output
0,1,-,1,0,0,1,-,-,-,1,0,0,0,0,2,-,3,-	0,1,X,1,0,0,1,X,-,-,X,1,0,0,0,0,2,X,3,-
0,1,1,1,1,1,2,3,-,3,1,1,0,1,1,1,2,-,-,-	0,1,1,1,1,1,2,3,X,3,1,1,0,1,1,1,2,X,-,X
0,0,0,1,-,-,3,3,-,2,0,0,1,2,-,-,-,2,3,-	0,0,0,1,-,X,3,3,X,2,0,0,1,2,X,-,-,2,3,-
0,1,2,4,-,-,-,2,1,2,1,1,1,-,2,-,1,-,-,-	0,1,2,4,X,X,X,2,1,2,1,1,1,X,2,-,1,-,X,-
0,1,-,-,-,3,2,-,2,-,2,1,2,-,2,-,2,1,1	0,1,X,X,X,-,3,2,-,2,X,2,1,2,-,2,X,2,1,1
0,1,2,3,3,-,-,2,1,2,-,2,0,1,-,2,1,1,0,0	0,1,2,3,3,-,X,2,1,2,X,2,0,1,X,2,1,1,0,0
0,0,0,0,1,-,-,-,1,1,2,3,2,2,2,2,1,0,0,0	0,0,0,0,1,X,-,X,1,1,2,3,2,2,2,2,1,0,0,0
1,1,1,1,2,-,-,2,1,0,1,-,-,-,-,1,0,1,1	1,1,1,1,2,-,X,2,1,0,1,X,X,-,-,X,1,0,1,1
-,-,-,-,2,1,-,2,2,3,-,4,-,-,1,1,0,2,-	-,X,-,-,X,2,1,-,2,2,3,-,4,X,-,1,1,0,2,X
-,-,2,1,1,1,2,2,3,-,-,-,2,1,2,1,1,0,2,-	X,2,1,1,1,2,2,3,X,X,-,X,2,1,2,1,1,0,2,X
1,-,1,1,0,1,-,-,-,3,1,1,0,2,-,3,2,2,2	1,-,1,1,0,1,X,X,-,-,3,1,1,0,2,X,3,2,2,2
1,2,-,1,0,1,-,2,-,-,2,0,0,0,2,-,-,-,-	1,2,X,1,0,1,-,2,-,X,2,0,0,0,2,X,X,-,X,-
2,-,4,3,2,2,1,1,2,-,3,2,1,1,1,2,-,-,1,1	2,X,4,3,2,2,1,1,2,X,3,2,1,1,1,2,-,-,1,1
-,-,4,-,-,2,-,1,1,3,-,4,-,2,0,1,-,2,1,0	-,X,4,X,X,2,X,1,1,3,X,4,X,2,0,1,-,2,1,0
-,-,4,4,3,-,-,1,0,2,-,-,-,2,0,1,-,-,1,0	-,X,4,4,3,-,-,1,0,2,X,-,X,2,0,1,X,X,1,0
-,-,-,-,-,1,1,0,1,2,-,-,2,1,2,3,-,2,1	-,-,X,-,X,-,1,1,0,1,2,X,-,2,1,2,3,-,2,1
1,-,2,2,-,-,-,2,1,1,-,2,-,2,3,-,-,3,-,-	1,X,2,2,-,-,X,2,1,1,-,2,X,2,3,X,-,3,X,-
1,1,1,1,2,3,-,-,-,2,1,1,1,-,-,-,3,-,-	1,1,1,1,2,3,-,-,X,2,1,1,1,-,X,X,X,3,X,-
0,0,0,2,-,-,1,1,2,-,1,0,1,-,-,4,2,3,-,2	0,0,0,2,X,X,1,1,2,X,1,0,1,-,X,4,2,3,-,2
0,0,0,2,-,3,1,-,1,-,1,0,1,-,-,1,0,1,-,1	0,0,0,2,X,3,1,-,1,-,1,0,1,X,-,1,0,1,X,1

Input Pysat

<pre> 0,1,X,1,0,0,1,X,-,-,X,1,0,0,0,0,2,X,3,- 0,1,1,1,1,1,2,3,X,3,1,1,0,1,1,1,2,X,-,- 0,0,0,1,-,X,3,3,X,2,0,0,1,2,X,-,-,2,3,- 0,1,2,4,X,X,X,2,1,2,1,1,1,X,2,-,1,-,X,- 0,1,X,X,X,-,3,2,-,2,X,2,1,2,-,2,X,2,1,1 0,1,2,3,3,-,X,2,1,2,X,2,0,1,X,2,1,1,0,0 0,0,0,0,1,X,-,X,1,1,2,3,2,2,2,2,1,0,0,0 1,1,1,1,2,-,X,2,1,0,1,X,X,-,-,X,1,0,1,1 -,X,-,-,X,2,1,-,2,2,3,-,4,X,-,1,1,0,2,X X,2,1,1,1,2,2,3,X,X,-,X,2,1,2,1,1,0,2,X 1,-,1,1,0,1,X,X,-,-,3,1,1,0,2,X,3,2,2,2 1,2,X,1,0,1,-,2,-,X,2,0,0,0,2,X,-,-,-,- 2,X,4,3,2,2,1,1,2,X,3,2,1,1,1,2,-,-,1,1 -,X,4,X,X,2,X,1,1,3,X,4,X,2,0,1,-,2,1,0 -,X,4,4,3,-,-,1,0,2,X,-,X,2,0,1,-,-,1,0 -,X,-,X,-,1,1,0,1,2,X,-,2,1,2,3,-,2,1 1,X,2,2,-,-,X,2,1,1,-,2,X,2,3,X,-,3,-,- 1,1,1,1,2,3,-,-,X,2,1,1,1,-,X,X,X,3,X,- 0,0,0,2,X,X,1,1,2,X,1,0,1,-,-,4,2,3,-,2 0,0,0,2,X,3,1,-,1,-,1,0,1,X,-,1,0,1,X,1 </pre>	<pre> 0,1,X,1,0,0,1,X,-,-,X,1,0,0,0,0,2,X,3,- 0,1,1,1,1,1,2,3,X,3,1,1,0,1,1,1,2,X,-,- 0,0,0,1,-,X,3,3,X,2,0,0,1,2,X,-,-,2,3,- 0,1,2,4,X,X,X,2,1,2,1,1,1,X,2,-,1,-,X,- 0,1,X,X,X,-,3,2,-,2,X,2,1,2,-,2,X,2,1,1 0,1,2,3,3,-,X,2,1,2,X,2,0,1,X,2,1,1,0,0 0,0,0,0,1,X,-,X,1,1,2,3,2,2,2,2,1,0,0,0 1,1,1,1,2,-,X,2,1,0,1,X,X,-,-,X,1,0,1,1 -,X,-,-,X,2,1,-,2,2,3,-,4,X,-,1,1,0,2,X X,2,1,1,1,2,2,3,X,X,-,X,2,1,2,1,1,0,2,X 1,-,1,1,0,1,X,X,-,-,3,1,1,0,2,X,3,2,2,2 1,2,X,1,0,1,-,2,-,X,2,0,0,0,2,X,-,-,-,- 2,X,4,3,2,2,1,1,2,X,3,2,1,1,1,2,-,-,1,1 -,X,4,X,X,2,X,1,1,3,X,4,X,2,0,1,-,2,1,0 -,X,4,4,3,-,-,1,0,2,X,-,X,2,0,1,-,-,1,0 -,X,-,-,-,-,1,1,0,1,2,X,-,2,1,2,3,-,2,1 1,X,2,2,-,-,X,2,1,1,-,2,X,2,3,X,-,3,-,- 1,1,1,1,2,3,-,-,X,2,1,1,1,-,X,X,X,3,-,- 0,0,0,2,X,X,1,1,2,X,1,0,1,-,-,4,2,3,-,2 0,0,0,2,X,3,1,-,1,-,1,0,1,X,-,1,0,1,-,1 </pre>
Resolution	Backtracking

Thời gian và bộ nhớ sau khi thực thi

STT	Thuật toán	Thời gian thực thi (s)				
		5x5	9x9	11x11	15x15	20x20
1	Resolution	0.005	0.051	0.189	0.2465	0.736
2	Backtracking	0.019	0.053	0.187	0.2449	0.742
3	Brute-force	0.003	Intractable	Intractable	Intractable	Intractable

Nhận xét

Về thời gian thực thi: Thuật toán resolution và backtracking có độ lệch không đáng kể, chạy trong khoảng thời gian rất ngắn. Đối với brute-force, kể từ mảng 9x9 trở lên, brute-force chạy vượt quá ngưỡng thời gian cho phép (8 giờ) nên không thể kết luận thời gian chính xác

Về độ chính xác (tất cả thuật toán so với pysat):

Pysat luôn luôn thực hiện chính xác nhất khi đưa đầy đủ các mệnh đề CNFs. Và do thuật toán pysat có cơ chế tự dò những mìn ngẫu nhiên nếu số mìn xung quanh một ô chưa bằng giá trị của ô đó.

VD: một trường hợp nhỏ của mảng 20x20, nằm góc phải trên cùng của input

	X	3	-			X	3	-	
	X	-	X			X	-	-	
	2	3	-			2	3	-	
	X	-	X			X	-	X	
Kết quả pysat					Kết quả resolution và backtracking				

Ở đây ta thấy được có 2 cách để giải:

Cách thứ nhất là đặt bom ở 2 ô (0,2) và (2,2)

Cách thứ hai là đặt bom ở 1 ô (1,2)

Pysat.solver sẽ chọn ngẫu nhiên trường hợp tối ưu để đặt bom (trường hợp 2). Các thuật toán do nhóm thiết kế không có cơ chế suy luận ngẫu nhiên như vậy nên kết quả không chính xác tuyệt đối bằng pysat). Do thuật toán khi gặp 2 trường hợp như vậy không biết xử lý trường hợp nào nên không đủ dữ kiện đưa ra kết quả.

Thêm vào đó, dựa vào những kết quả khi thực thi các thuật toán cho thấy rằng resolution tốt hơn backtracking do ít kết quả sai lệch với pysat nhất.

6. Phân công công việc

STT	MSSV	Họ và tên	Nhiệm vụ phân công	Mức độ hoàn thành
1	21127623	Hà Minh Khang	Thực thi xử lý CNF Viết báo cáo	100%
2	21127684	Kuo Yung Sheng	Thực thi thuật toán dpll (backtracking) Thực thi thuật toán resolution	100%
3	21127724	Nguyễn Đình Vinh	Thực thi thuật toán brute-force Khởi tạo các bộ test case Quay video demo	100%

7. Đánh giá mức độ hoàn thành

STT	Công việc	Mức độ hoàn thành
1	Mô tả chính xác nguyên tắc logic để khởi tạo các mệnh đề CNFs	100%
2	Khởi tạo các mệnh đề CNFs	100%
3	Áp dụng pysat để giải CNF chính xác	100%
4	Áp dụng resolution để giải CNF	100%
5	Áp dụng brute-force để giải CNF	100%
6	Áp dụng backtracking để giải CNF	100%
7	Viết báo cáo (phân tích, thực nghiệm, test case, so sánh)	100%
8	Quay video demo chương trình	100%

8. TÀI LIỆU THAM KHẢO

<https://people.eecs.berkeley.edu/~daw/teaching/cs70-s05/Notes/lecture09.pdf>

<https://courses.cs.duke.edu/cps102/spring05/notes/minesweeper.pdf>

https://en.wikipedia.org/wiki/DPLL_algorithm

<https://pysathq.github.io/docs/html/api/solvers.html>