Trường Đại học Khoa học Tự nhiên – ĐHQG TP.HCM

Môn học

Cơ sở trí tuệ nhân tạo

Báo cáo đồ án

Project 2 – Hashiwokakero

Giảng viên phụ trách:

Tiến Sĩ Bùi Duy Đăng

Thầy Huỳnh Lâm Hải Đăng

Thầy Lê Nhựt Nam

TP Hồ Chí Minh, tháng 4 năm 2025

Phụ lục:

I. THÔNG TIN CHUNG:

1. Thông tin thành viên:

- Khoa: Công nghệ Thông tin (CLC)
- Lớp: 23CLC09
- Thành viên của nhóm:
  - Nguyễn Hữu Kiến Phi – 23127242
  - Lê Hoàng Long – 23127412
  - Lê Nguyễn Nhật Khánh – 23127206
  - Lê Trọng Đạt – 23127169

2. Thông tin đồ án:

- Tên đồ án: Hashiwokakero (Project 2)
- Tổng quan về đồ án:
  - Hashiwokakero là một câu đố logic thách thức người chơi kết nối các đảo được đánh số theo các quy tắc đơn giản.
  - Được tạo ra bởi Nikoli, dần trở nên phổ biến hơn bởi sự hấp dẫn, tính thử thách và chiều sâu logic của nó.
  - Câu đố được tổ chức trên một lưới hình chữ nhật, mỗi đảo được đánh số từ 1 đến 8 để biểu diễn cho số cầu cần kết nối cho mỗi đảo.
  - Trong đồ án này, chúng ta phải dùng Conjunctive Normal Form(CNF) logic để giải câu đố trên.
- Một số ràng buộc của câu đố:
  - Chúng phải bắt đầu và kết thúc ở các đảo riêng biệt và các cầu chỉ được xây dựng theo đường thẳng phương ngang hoặc phương thẳng đứng.
  - Các cây cầu không được cắt ngang nhau hay cắt qua đảo.
  - Mỗi cặp đảo chỉ được nối nhiều nhất là 2 cầu.
  - Các cây cầu phải nối các đảo thành một nhóm duy nhất.
- Các thuật toán yêu cầu:
  - Thuật toán tự động tạo các mệnh đề CNF.
  - Sử dụng module PySAT để giải CNFs.
  - Sử dụng A* để giải CNFs.
  - Sử dụng Brute-force để giải CNFs.
  - Sử dụng Backtracking để giải CNFs.

II. PHÂN CHIA CÔNG VIỆC:

THÀNH VIÊN	CÔNG VIỆC	MỨC ĐỘ HOÀN THÀNH
Nguyễn Hữu Kiến Phi	Viết chương trình tạo CNFs, dùng PySAT, tối ưu cho thuật toán A*, thuật toán backtracking.	100%
Lê Hoàng Long	Soạn test case, viết chương trình cho thuật toán Brute-force, quay video demo cho đồ án.	100%
Lê Nguyễn Nhật Khánh	Mô tả các nguyên tắc chính xác để tạo CNFs, viết report, readme, thử nghiệm chương trình.	100%
Lê Trọng Đạt	Viết chương trình cho thuật toán A*, bổ sung report, thử nghiệm chương trình.	100%

III. CHI TIẾT ĐỒ ÁN:

1. Tổng quan về test case:

- Test case có thể là hình chữ nhật hoặc hình vuông, các file input đã được sắp xếp từ dễ tới khó.
- Độ khó của test case càng cao thì thời gian xử lý để đưa ra kết quả cũng sẽ lâu hơn rất nhiều.
- Trong file input, với các số lớn hơn không thì sẽ biểu diễn cho các đảo, ngược lại là các số bằng không sẽ là các khoảng trống có thể đặt cầu.
- Test case nhỏ nhất sẽ là 3x3 và lớn nhất sẽ lên đến 20x20.

2. Một số lưu ý về định dạng của file output:

- Mỗi ô trong lưới sẽ được phân tách nhau bằng dấu phẩy.
- Các giá trị '0' là các vị trí trống.
- Các giá trị từ 1->8 dùng để biển diễn đảo.

- Một số kí tự đặc biệt như:
  - Kí hiệu '|' có nghĩa là 1 cầu thẳng đứng.
  - Kí hiệu '$' có nghĩa là 2 cầu thẳng đứng.
  - Kí hiệu '-' có nghĩa là 1 cầu thẳng ngang.
  - Kí hiệu '=' có nghĩa là 2 cầu thẳng ngang.

3. Mô tả các nguyên tắc logic chính xác để tạo CNF:

a) Giới thiệu về hệ thống CNFs:

- Trong hệ thống CNF, các mệnh đề logic được biểu diễn dưới dạng các clause. Mỗi clause là một phép OR giữa các biến logic, và toàn bộ hệ thống CNF là một phép AND giữa các clause.
- Ví dụ về CNF: $(A \vee B) \wedge (\neg A \vee C)$
- Sau khi có hệ thống CNF, ta có thể dùng module pysat hoặc các thuật toán để tìm ra kết quả.

b) Ràng buộc cho các đảo:

- Mỗi đảo trong trò chơi Hashiwokakero có một số cầu cụ thể mà nó cần phải kết nối với các đảo khác. Đây là thông tin cơ bản để tạo ra các ràng buộc.
- Các ràng buộc này phải được biểu diễn dưới dạng các biến logic, và số cầu cần thiết cho mỗi đảo được biểu diễn bằng một tập hợp các biến logic.
- Sau khi tạo xong, CNF lúc này sẽ chứa các điều kiện ràng buộc sao cho số cầu kết nối với mỗi đảo là đúng.

c) Ràng buộc cho các cầu nối giữa các đảo:

- Một đảo có thể kết nối với các đảo khác theo các hướng(ngang hoặc dọc) và mỗi kết nối có thể có một cầu đơn hoặc đôi.
- Khi ta tìm thấy một đảo khác có khả năng kết nối với đảo hiện tại thì ta sẽ tạo ra CNF mới biểu diễn sự liên kết này(cả 1 cầu và 2 cầu), sau đó thêm nó vào hệ thống CNF.
- Các clause sẽ đảm bảo việc kết nối giữa các đảo là đúng, không dư thừa cũng không thiếu.

d) Ràng buộc không giao nhau:

- Không thể có hai cầu giao nhau tại một vị trí khác đảo trong bài toán này.

- Các biến logic sẽ đảm bảo rằng hai cầu không cắt nhau, tức là hai cầu không thể chia sẻ cùng một ô ngoại trừ ô đảo.
- Điều này được thực hiện bằng cách kiểm tra các cặp cầu có khả năng giao nhau và tạo ra các điều kiện logic cấm giao nhau tại ô không phải đảo.

e) Biến logic và biểu diễn CNF:

- Các biến logic được đặt cho từng cầu giữa hai đảo. Các biến này có thể có giá trị True (có cầu) hoặc False (không có cầu).
- Những biến này sẽ được tổ hợp thành các clauses. Các clauses sẽ đảm bảo rằng số lượng cầu đúng với yêu cầu của mỗi đảo và rằng các cầu không giao nhau.

f) Giải CNF:

- Sau khi hình thành được CNF thông qua lớp CNFGenerator, khi này các điều kiện ràng buộc được hình thành và chúng ta có thể từ đó giải mã bằng PySAT hoặc A*. Khi tìm thấy một tổ hợp thỏa mãn thì chúng ta sẽ phải kiểm tra xem tổ hợp trên có tạo thành một lưới liên thông hay không, nếu không thì sẽ bỏ qua và tiếp tục tìm kiếm. Việc giải mã và mã hóa được thực hiện bằng lớp LogicVariable, giúp kiểm tra dễ dàng.

● Kết luận : Cách tổ chức như trên sẽ giúp việc xây dựng và kiểm tra hệ thống CNF trở nên rõ ràng, dễ mở rộng

4. Thuật toán tạo CNFs:

a) Giới thiệu:

- Là quá trình tạo ra hệ thống CNF bằng các nguyên tắc logic chính xác ở trên.
- Sau khi tạo được CNFs, chúng ta có thể dễ dàng giải quyết được yêu cầu của bài toán.

b) Chi tiết:

- Đầu tiên chúng ta sẽ duyệt từ đầu đến cuối lưới, nếu tìm thấy nó là một đảo thì chúng ta tiến hành tới bước tiếp theo. Chúng ta sẽ tìm theo 4 hướng để tìm các đảo có khả năng nối cầu với nó và thêm vào hệ thống CNF.
- Sau đó chúng ta sẽ tiến hành loại bỏ các tổ hợp vượt quá số cầu theo các chỉ số của đảo.

- Và cuối cùng chúng ta sẽ xét những trường hợp cầu cắt nhau và thêm các ràng buộc này vào hệ thống CNF.
- Khi có được hệ thống CNF, chúng ta có thể dễ dàng giải quyết nó bằng các thuật toán như A*, backtracking,...

5. Sử dụng module PySAT để giải các CNFs:

a) Giới thiệu:

- PySAT(Python Boolean Satisfiability Problem Solver Library) là một thư viện mạnh mẽ giúp giải các bài toán thỏa mãn mệnh đề (SAT - Boolean Satisfiability Problem) bằng cách sử dụng các bộ giải SAT hiện đại như MiniSat, Glucose, Lingeling,...

b) Chi tiết:

- Sau khi có được hệ thống CNF, thì chúng ta sẽ sử dụng Glucose của pysat để giải và nhận được đáp án cuối cùng.
- Sau đó chúng ta chuyển đổi nó sang dạng mảng 2 chiều và ghi kết quả vào file output.

6. Thuật toán A Star (A*):

a) Giới thiệu:

- Đây là một thuật toán dựa trên sự đánh giá heuristic (có thể gọi là hàm ước lượng) để tìm kiếm, được ưa chuộng cho việc tìm đường đi. Sử dụng hàng đợi ưu tiên để lưu trữ dữ liệu. Ở bài này thì heuristic sẽ được tính dựa trên CNFs (mệnh đề logic).

b) Chi tiết:

- Heuristic function: f(x) = g(x) + h(x). Trong đó g(x) là số bước đã đi từ trạng thái ban đầu, h(x) là ‘score’
- Hàm mở rộng trạng thái sẽ gán “true” hoặc “false” cho biến chưa được gán.
- ‘score’ được tính bằng hàm heuristic. Đầu tiên sẽ duyệt qua các mệnh đề, nếu mệnh đề được thỏa mãn thì bỏ qua. Đối với mệnh đề chưa thỏa mãn, tìm các biến chưa được gán, hoặc các biến đã gán sai và cộng điểm ‘score’. Nếu không còn biến nào chưa gán thì +50, có đúng 1 biến chưa gán +8, có đúng 2 biến chưa gán +4, nhiều hơn 2 biến chưa gán thì +1.
- Lí do có sự chênh lệch khác nhau vậy là khi một mệnh đề không được thỏa mãn tức là nó ở trạng thái rất xấu (gần như không thể thỏa mãn).

+50 sẽ giúp tránh xa trạng thái này, nếu không sẽ lặp vô tận để mở rộng những trạng thái vô vọng. Tương tự với +8, +4, +1.

- Khởi tạo trạng thái, hàng đợi ưu tiên. Đưa các trạng thái vào hàng đợi ưu tiên và bắt đầu giải. Lấy trạng thái tốt nhất từ hàng đợi (dựa trên heuristic), nếu đạt đích thì trả về kết quả, không thì tiếp tục mở rộng và thêm tiếp vào hàng đợi.

7. Thuật toán Brute-force:

a) Giới thiệu:

- Brute-force (hay tìm kiếm vét cạn) là một phương pháp thử tất cả các khả năng có thể để tìm ra lời giải đúng.
- Thuật toán sẽ liệt kê tất cả trường hợp và thử để tìm ra kết quả cuối cùng.
- Ưu điểm là nó sẽ luôn tìm ra được kết quả. Tuy nhiên vì không có tính tối ưu nên hầu hết các trường hợp đều cho thời gian phản hồi khá lâu.
- Tốn tài nguyên bộ nhớ và CPU

b) Chi tiết:

- Tạo ra một dictionary để chứa tất cả các biến logic mà ta sẽ thử.
- Sau đó sẽ dùng backtracking để duyệt qua tất cả các trường hợp. Trong mỗi trường hợp chúng ta sẽ kiểm tra xem model hiện tại đã thỏa hết tất cả các clause chưa. Nếu như nó thỏa mãn tất cả các CNFs thì sẽ trả về model hiện tại.

8. Thuật toán Backtracking:

a) Giới thiệu:

- Ở đây chúng ta sử dụng thuật toán backtracking có pruning để tối ưu hóa tài nguyên và thời gian chạy.
- Chúng ta sẽ thử các giá trị của các biến logic (biến trong hệ CNF) cho đến khi tìm được một mô hình thỏa mãn tất cả các ràng buộc. Việc sử dụng cắt tỉa (pruning) giúp loại bỏ những trường hợp không cần thiết trong quá trình tìm kiếm.

b) Chi tiết:

- Tạo ra một dictionary chứa tất cả các biến logic mà ta sẽ thử.
- Dùng backtracking để có thể thử hết tất cả trường hợp.

- Đồng thời chúng ta cũng sẽ có một hàm dùng để kiểm tra model hiện tại là có thỏa mãn hay không. Ngoài ra hàm này còn có chức năng pruning như sau:
  - Đối với mỗi clause, hàm kiểm tra tất cả các literal trong clause đó. Nếu một literal đã được gán giá trị và thỏa mãn clause, thì biến kiểm tra được đặt thành True. Nếu không, nó sẽ kiểm tra xem còn literal nào chưa được quyết định giá trị không.
  - Nếu một clause không thỏa mãn và tất cả các literal trong clause đều đã được quyết định giá trị, thì mô hình không hợp lệ và hàm trả về False.

9. Thử nghiệm test case và đưa ra so sánh giữa các thuật toán:

a) Test case 1 - input-04(5x6):

- Sử dụng module pysat:

```
Loaded grid:
[3, 0, 5, 0, 0, 2]
[0, 0, 0, 0, 0, 0]
[6, 0, 8, 0, 0, 3]
[0, 0, 0, 0, 0, 0]
[4, 0, 4, 0, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[✓] Tìm được lời giải lần 1, kiểm tra liên thông...
[✓] Lời giải liên thông hợp lệ.
[✓] PySAT chạy trong 0.0009 giây

Solution found:
3 , - , 5 , = , = , 2
$ , 0 , $ , 0 , 0 , 0
6 , = , 8 , = , = , 3
$ , 0 , $ , 0 , 0 , |
4 , = , 4 , 0 , 0 , 1
```

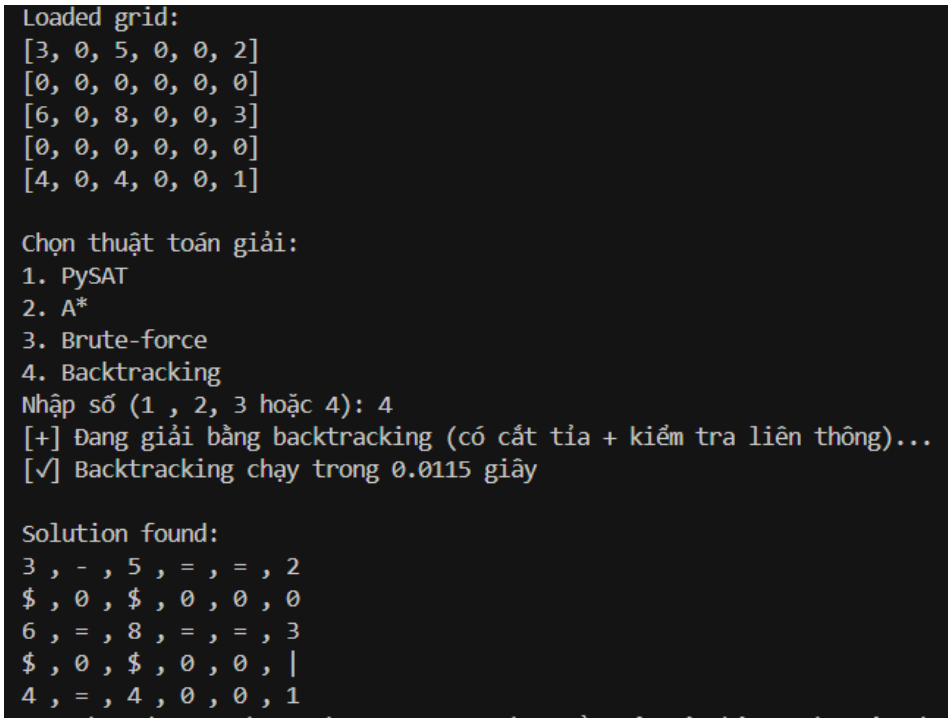
- Sử dụng A*:

```
Loaded grid:
[3, 0, 5, 0, 0, 2]
[0, 0, 0, 0, 0, 0]
[6, 0, 8, 0, 0, 3]
[0, 0, 0, 0, 0, 0]
[4, 0, 4, 0, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 2
[+] Đang giải bằng A* trên CNF (logic-level)...
[✓] Lời giải liên thông hợp lệ.
[✓] A* chạy trong 0.0701 giây

Solution found:
3 , - , 5 , = , = , 2
$ , 0 , $ , 0 , 0 , 0
6 , = , 8 , = , = , 3
$ , 0 , $ , 0 , 0 , |
4 , = , 4 , 0 , 0 , 1
```

- Sử dụng Brute-force:

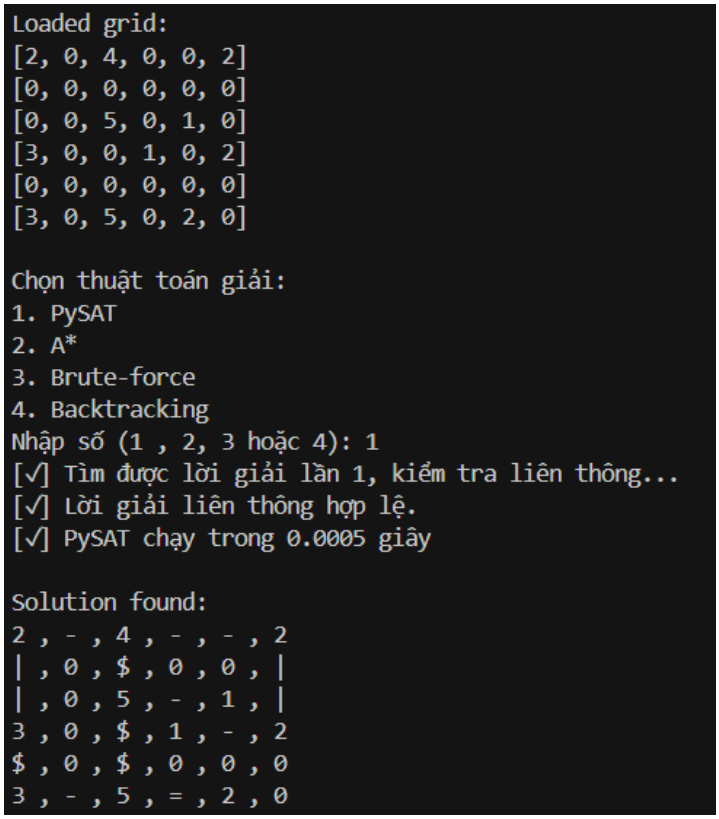
```
Loaded grid:
[3, 0, 5, 0, 0, 2]
[0, 0, 0, 0, 0, 0]
[6, 0, 8, 0, 0, 3]
[0, 0, 0, 0, 0, 0]
[4, 0, 4, 0, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 3
[+] Đang giải bằng brute-force...
[✓] Brute-force chạy trong 39.1316 giây

Solution found:
3 , - , 5 , = , = , 2
$ , 0 , $ , 0 , 0 , 0
6 , = , 8 , = , = , 3
$ , 0 , $ , 0 , 0 , |
4 , = , 4 , 0 , 0 , 1
```

- Sử dụng Backtracking:

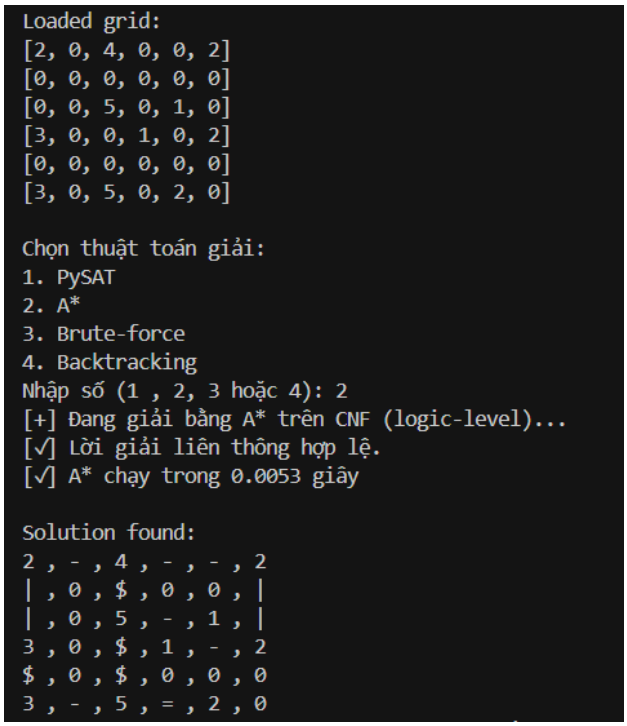


b) Test case 2 - input-05(6x6):

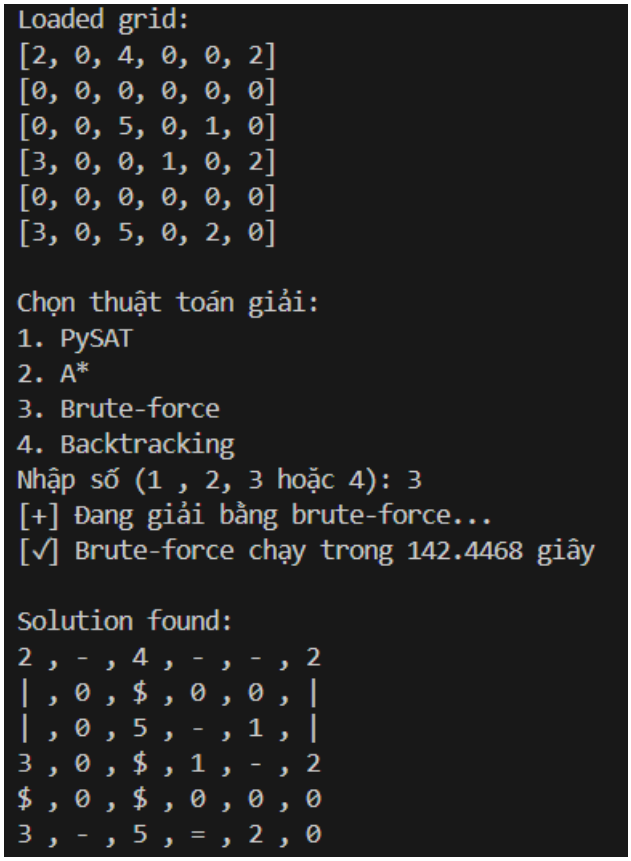
- Sử dụng module pysat:



- Sử dụng A*:



- Sử dụng Brute-force:



- Sử dụng Backtracking:

```
Loaded grid:
[2, 0, 4, 0, 0, 2]
[0, 0, 0, 0, 0, 0]
[0, 0, 5, 0, 1, 0]
[3, 0, 0, 1, 0, 2]
[0, 0, 0, 0, 0, 0]
[3, 0, 5, 0, 2, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[√] Backtracking chạy trong 0.0045 giây

Solution found:
2 , - , 4 , - , - , 2
| , 0 , $ , 0 , 0 , |
| , 0 , 5 , - , 1 , |
3 , 0 , $ , 1 , - , 2
$ , 0 , $ , 0 , 0 , 0
3 , - , 5 , = , 2 , 0
```

c) Test case 3 - input-06(7x7):

- Sử dụng module pysat:

```
Loaded grid:
[3, 0, 3, 0, 0, 2, 0]
[0, 1, 0, 3, 0, 0, 0]
[3, 0, 1, 0, 0, 0, 1]
[0, 3, 0, 4, 0, 2, 0]
[1, 0, 1, 0, 3, 0, 4]
[0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 2, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[√] Tìm được lời giải lần 1, kiểm tra liên thông...
[√] Lời giải liên thông hợp lệ.
[√] PySAT chạy trong 0.0008 giây

Solution found:
3 , = , 3 , - , - , 2 , 0
| , 1 , - , 3 , 0 , | , 0
3 , - , 1 , $ , 0 , | , 1
| , 3 , - , 4 , - , 2 , |
1 , $ , 1 , - , 3 , = , 4
0 , $ , 0 , 0 , 0 , 0 , |
0 , 3 , - , 2 , - , - , 2
```

- Sử dụng A*:

```
Loaded grid:
[3, 0, 3, 0, 0, 2, 0]
[0, 1, 0, 3, 0, 0, 0]
[3, 0, 1, 0, 0, 0, 1]
[0, 3, 0, 4, 0, 2, 0]
[1, 0, 1, 0, 3, 0, 4]
[0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 2, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 2
[+] Đang giải bằng A* trên CNF (logic-level)...
[✓] Lời giải liên thông hợp lệ.
[✓] A* chạy trong 0.7218 giây

Solution found:
3 , = , 3 , - , - , 2 , 0
| , 1 , - , 3 , 0 , | , 0
3 , - , 1 , $ , 0 , | , 1
| , 3 , - , 4 , - , 2 , |
1 , $ , 1 , - , 3 , = , 4
0 , $ , 0 , 0 , 0 , 0 , |
0 , 3 , - , 2 , - , - , 2
```

- Sử dụng Backtracking:

```
Loaded grid:
[3, 0, 3, 0, 0, 2, 0]
[0, 1, 0, 3, 0, 0, 0]
[3, 0, 1, 0, 0, 0, 1]
[0, 3, 0, 4, 0, 2, 0]
[1, 0, 1, 0, 3, 0, 4]
[0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 2, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[✓] Backtracking chạy trong 0.0173 giây

Solution found:
3 , = , 3 , - , - , 2 , 0
| , 1 , - , 3 , 0 , | , 0
3 , - , 1 , $ , 0 , | , 1
| , 3 , - , 4 , - , 2 , |
1 , $ , 1 , - , 3 , = , 4
0 , $ , 0 , 0 , 0 , 0 , |
0 , 3 , - , 2 , - , - , 2
```

d) Test case 4 - input-08(9x9):

- Sử dụng module pysat:

```
Loaded grid:
[0, 1, 0, 0, 2, 0, 2, 0, 2]
[2, 0, 2, 0, 0, 1, 0, 2, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 0, 3, 0, 4, 0, 0, 3, 0]
[3, 0, 0, 2, 0, 2, 0, 0, 3]
[0, 2, 0, 0, 2, 0, 2, 0, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 4, 0, 0, 4, 0, 2]
[2, 0, 2, 0, 0, 2, 0, 1, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[√] Tìm được lời giải lần 1, kiểm tra liên thông...
[√] Lời giải liên thông hợp lệ.
[√] PySAT chạy trong 0.0016 giây
```

```
Solution found:
0 , 1 , - , - , 2 , - , 2 , - , 2
2 , - , 2 , 0 , 0 , 1 , - , 2 , |
| , 0 , | , 0 , 0 , 0 , 0 , | , 2
| , 0 , 3 , = , 4 , = , = , 3 , |
3 , - , - , 2 , - , 2 , - , - , 3
| , 2 , - , - , 2 , - , 2 , 0 , |
2 , | , 0 , 0 , 0 , 0 , | , 0 , |
| , 3 , = , 4 , = , = , 4 , - , 2
2 , - , 2 , - , - , 2 , - , 1 , 0
```

- Sử dụng A*:

```
Loaded grid:
[0, 1, 0, 0, 2, 0, 2, 0, 2]
[2, 0, 2, 0, 0, 1, 0, 2, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 0, 3, 0, 4, 0, 0, 3, 0]
[3, 0, 0, 2, 0, 2, 0, 0, 3]
[0, 2, 0, 0, 2, 0, 2, 0, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 4, 0, 0, 4, 0, 2]
[2, 0, 2, 0, 0, 2, 0, 1, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 2
[+] Đang giải bằng A* trên CNF (logic-level)...
[x] Lời giải 1 không liên thông, thử lại...
[x] Lời giải 2 không liên thông, thử lại...
[√] Lời giải liên thông hợp lệ.
[√] A* chạy trong 60.6418 giây
```

```
Solution found:
0 , 1 , - , - , 2 , - , 2 , - , 2
2 , - , 2 , 0 , 0 , 1 , - , 2 , |
| , 0 , | , 0 , 0 , 0 , 0 , | , 2
| , 0 , 3 , = , 4 , = , = , 3 , |
3 , - , - , 2 , - , 2 , - , - , 3
| , 2 , - , - , 2 , - , 2 , 0 , |
2 , | , 0 , 0 , 0 , 0 , | , 0 , |
| , 3 , = , 4 , = , = , 4 , - , 2
2 , - , 2 , - , - , 2 , - , 1 , 0
```

- Sử dụng Backtracking:

```
Loaded grid:
[0, 1, 0, 0, 2, 0, 2, 0, 2]
[2, 0, 2, 0, 0, 1, 0, 2, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 0, 3, 0, 4, 0, 0, 3, 0]
[3, 0, 0, 2, 0, 2, 0, 0, 3]
[0, 2, 0, 0, 2, 0, 2, 0, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 3, 0, 4, 0, 0, 4, 0, 2]
[2, 0, 2, 0, 0, 2, 0, 1, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[√] Backtracking chạy trong 2.1496 giây
```

```
Solution found:
0 , 1 , - , - , 2 , - , 2 , - , 2
2 , - , 2 , 0 , 0 , 1 , - , 2 , |
| , 0 , | , 0 , 0 , 0 , 0 , | , 2
| , 0 , 3 , = , 4 , = , = , 3 , |
3 , - , - , 2 , - , 2 , - , - , 3
| , 2 , - , - , 2 , - , 2 , 0 , |
2 , | , 0 , 0 , 0 , 0 , | , 0 , |
| , 3 , = , 4 , = , = , 4 , - , 2
2 , - , 2 , - , - , 2 , - , 1 , 0
```

e) <u>Test case 5 - input-10(11x11):</u>

- Sử dụng module pysat:

```
Loaded grid:
[1, 0, 0, 1, 0, 0, 2, 0, 1, 0, 1]
[0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[2, 0, 1, 0, 2, 0, 4, 0, 5, 0, 4]
[0, 2, 0, 4, 0, 2, 0, 0, 0, 1, 0]
[3, 0, 1, 0, 2, 0, 0, 4, 0, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[3, 0, 0, 5, 0, 0, 1, 0, 1, 0, 3]
[0, 0, 1, 0, 2, 0, 0, 5, 0, 2, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
[2, 0, 3, 0, 4, 0, 0, 5, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[√] Tìm được lời giải lần 1, kiểm tra liên thông...
[√] Lời giải liên thông hợp lệ.
[√] PySAT chạy trong 0.0027 giây
```

```
Solution found:
1 , 0 , 0 , 1 , - , - , 2 , 0 , 1 , 0 , 1
| , 2 , - , - , - , 1 , | , 0 , | , 0 , |
2 , | , 1 , - , 2 , - , 4 , = , 5 , = , 4
| , 2 , - , 4 , - , 2 , - , - , - , 1 , |
3 , - , 1 , $ , 2 , = , = , 4 , - , - , 3
| , 0 , 0 , $ , 0 , 0 , 0 , | , 0 , 0 , |
3 , - , - , 5 , - , - , 1 , | , 1 , - , 3
| , 0 , 1 , | , 2 , - , - , 5 , - , 2 , |
2 , 0 , | , | , | , 0 , 0 , $ , 0 , | , 2
| , 0 , | , 1 , | , 0 , 0 , $ , 0 , 1 , |
2 , - , 3 , - , 4 , = , = , 5 , - , - , 2
```

- Sử dụng Backtracking:

```
Loaded grid:
[1, 0, 0, 1, 0, 0, 2, 0, 1, 0, 1]
[0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0]
[2, 0, 1, 0, 2, 0, 4, 0, 5, 0, 4]
[0, 2, 0, 4, 0, 2, 0, 0, 0, 1, 0]
[3, 0, 1, 0, 2, 0, 0, 4, 0, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[3, 0, 0, 5, 0, 0, 1, 0, 1, 0, 3]
[0, 0, 1, 0, 2, 0, 0, 5, 0, 2, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0]
[2, 0, 3, 0, 4, 0, 0, 5, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[√] Backtracking chạy trong 1.1572 giây
```

```
Solution found:
1 , 0 , 0 , 1 , - , - , 2 , 0 , 1 , 0 , 1
| , 2 , - , - , - , 1 , | , 0 , | , 0 , |
2 , | , 1 , - , 2 , - , 4 , = , 5 , = , 4
| , 2 , - , 4 , - , 2 , - , - , - , 1 , |
3 , - , 1 , $ , 2 , = , = , 4 , - , - , 3
| , 0 , 0 , $ , 0 , 0 , 0 , | , 0 , 0 , |
3 , - , - , 5 , - , - , 1 , | , 1 , - , 3
| , 0 , 1 , | , 2 , - , - , 5 , - , 2 , |
2 , 0 , | , | , | , 0 , 0 , $ , 0 , | , 2
| , 0 , | , 1 , | , 0 , 0 , $ , 0 , 1 , |
2 , - , 3 , - , 4 , = , = , 5 , - , - , 2
```

f) Test case 6 - input-11(11x11):

- Sử dụng module pysat:

```
Loaded grid:
[1, 0, 2, 0, 2, 0, 3, 0, 1, 0, 2]
[0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0]
[1, 0, 4, 0, 7, 0, 5, 0, 1, 0, 4]
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[3, 0, 2, 0, 0, 0, 0, 3, 0, 0, 5]
[0, 0, 0, 0, 4, 0, 4, 0, 0, 1, 0]
[2, 0, 0, 4, 0, 2, 0, 0, 4, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[2, 0, 1, 0, 1, 0, 2, 0, 6, 0, 4]
[0, 1, 0, 3, 0, 2, 0, 1, 0, 0, 0]
[2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[√] Tìm được lời giải lần 1, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[√] Tìm được lời giải lần 2, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[√] Tìm được lời giải lần 3, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[√] Tìm được lời giải lần 4, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[√] Tìm được lời giải lần 5, kiểm tra liên thông...
[√] Lời giải liên thông hợp lệ.
[√] PySAT chạy trong 0.0060 giây
```

```
Solution found:
1 , - , 2 , 0 , 2 , - , 3 , 0 , 1 , - , 2
0 , 0 , | , 0 , | , 0 , $ , 3 , = , 2 , |
1 , 0 , 4 , = , 7 , = , 5 , | , 1 , - , 4
| , 0 , | , 1 , $ , 0 , | , | , 0 , 0 , $
3 , - , 2 , | , $ , 0 , | , 3 , = , = , 5
| , 0 , 0 , | , 4 , = , 4 , - , - , 1 , |
2 , - , - , 4 , - , 2 , - , - , 4 , - , 3
0 , 0 , 0 , | , 0 , 0 , 0 , 0 , $ , 0 , |
2 , - , 1 , | , 1 , - , 2 , - , 6 , = , 4
| , 1 , - , 3 , - , 2 , - , 1 , | , 0 , |
2 , - , 2 , - , 2 , - , 2 , - , 2 , 0 , 1
```

- Sử dụng Backtracking:

```
Loaded grid:
[1, 0, 2, 0, 2, 0, 3, 0, 1, 0, 2]
[0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0]
[1, 0, 4, 0, 7, 0, 5, 0, 1, 0, 4]
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[3, 0, 2, 0, 0, 0, 0, 3, 0, 0, 5]
[0, 0, 0, 0, 4, 0, 4, 0, 0, 1, 0]
[2, 0, 0, 4, 0, 2, 0, 0, 4, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[2, 0, 1, 0, 1, 0, 2, 0, 6, 0, 4]
[0, 1, 0, 3, 0, 2, 0, 1, 0, 0, 0]
[2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[✓] Backtracking chạy trong 38.8936 giây
```

```
Solution found:
1 , - , 2 , 0 , 2 , - , 3 , 0 , 1 , - , 2
0 , 0 , | , 0 , | , 0 , $ , 3 , = , 2 , |
1 , 0 , 4 , = , 7 , = , 5 , | , 1 , - , 4
| , 0 , | , 1 , $ , 0 , | , | , 0 , 0 , $
3 , - , 2 , | , $ , 0 , | , 3 , = , = , 5
| , 0 , 0 , | , 4 , = , 4 , - , - , 1 , |
2 , - , - , 4 , - , 2 , - , - , 4 , - , 3
0 , 0 , 0 , | , 0 , 0 , 0 , 0 , $ , 0 , |
2 , - , 1 , | , 1 , - , 2 , - , 6 , = , 4
| , 1 , - , 3 , - , 2 , - , 1 , | , 0 , |
2 , - , 2 , - , 2 , - , 2 , - , 2 , 0 , 1
```

g) Test case 7 - input-12(13x13):

- Sử dụng module pysat:

```
Loaded grid:
[2, 0, 2, 0, 0, 2, 0, 1, 0, 3, 0, 2, 0]
[0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 1]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 3, 0]
[3, 0, 0, 2, 0, 6, 0, 3, 0, 0, 2, 0, 2]
[0, 3, 0, 0, 1, 0, 0, 0, 0, 1, 0, 3, 0]
[3, 0, 0, 3, 0, 7, 0, 3, 0, 0, 2, 0, 2]
[0, 2, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0, 2, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 3]
[0, 4, 0, 5, 0, 5, 0, 0, 2, 0, 0, 0, 0, 0]
[2, 0, 0, 0, 1, 0, 0, 3, 0, 5, 0, 2, 0]
[0, 5, 0, 6, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2]
[0, 2, 0, 4, 0, 2, 0, 3, 0, 0, 0, 1, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[✓] Tìm được lời giải lần 1, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 2, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 3, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 4, kiểm tra liên thông...
[✓] Lời giải liên thông hợp lệ.
[✓] PySAT chạy trong 0.0044 giây
```

```
Solution found:
2 , - , 2 , - , - , 2 , 0 , 1 , - , 3 , - , 2 , 0
| , 2 , - , 2 , 0 , | , 0 , 0 , 0 , | , 1 , | , 1
| , | , 0 , | , 0 , | , 1 , - , - , 3 , | , 3 , |
3 , | , 0 , 2 , - , 6 , = , 3 , 0 , | , 2 , $ , 2
$ , 3 , - , - , 1 , $ , 0 , | , 0 , 1 , | , 3 , |
3 , | , 0 , 3 , = , 7 , - , 3 , - , - , 2 , | , 2
| , 2 , 0 , | , 0 , $ , 1 , - , 3 , - , - , 2 , |
2 , | , 0 , | , 0 , $ , 0 , 0 , | , 2 , - , - , 3
| , 4 , - , 5 , = , 5 , - , - , 2 , | , 0 , 0 , |
2 , $ , 0 , | , 1 , - , - , 3 , - , 5 , = , 2 , |
| , 5 , = , 6 , - , - , 1 , | , 0 , | , 0 , 0 , |
1 , | , 0 , $ , 0 , 0 , 0 , | , 0 , 2 , - , - , 2
0 , 2 , - , 4 , - , 2 , - , 3 , - , - , - , 1 , 0
```

- Sử dụng Backtracking:

```
Loaded grid:
[2, 0, 2, 0, 0, 2, 0, 1, 0, 3, 0, 2, 0]
[0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 1, 0, 1]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 3, 0]
[3, 0, 0, 2, 0, 6, 0, 3, 0, 0, 2, 0, 2]
[0, 3, 0, 0, 1, 0, 0, 0, 0, 1, 0, 3, 0]
[3, 0, 0, 3, 0, 7, 0, 3, 0, 0, 2, 0, 2]
[0, 2, 0, 0, 0, 0, 1, 0, 3, 0, 0, 2, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 3]
[0, 4, 0, 5, 0, 5, 0, 0, 2, 0, 0, 0, 0]
[2, 0, 0, 0, 1, 0, 0, 3, 0, 5, 0, 2, 0]
[0, 5, 0, 6, 0, 0, 1, 0, 0, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2]
[0, 2, 0, 4, 0, 2, 0, 3, 0, 0, 0, 1, 0]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[√] Backtracking chạy trong 6.7666 giây
```

```
Solution found:
2 , - , 2 , - , - , 2 , 0 , 1 , - , 3 , - , 2 , 0
| , 2 , - , 2 , 0 , | , 0 , 0 , 0 , | , 1 , | , 1
| , | , 0 , | , 0 , | , 1 , - , - , 3 , | , 3 , |
3 , | , 0 , 2 , - , 6 , = , 3 , 0 , | , 2 , $ , 2
$ , 3 , - , - , 1 , $ , 0 , | , 0 , 1 , | , 3 , |
3 , | , 0 , 3 , = , 7 , - , 3 , - , - , 2 , | , 2
| , 2 , 0 , | , 0 , $ , 1 , - , 3 , - , - , 2 , |
2 , | , 0 , | , 0 , $ , 0 , 0 , | , 2 , - , - , 3
| , 4 , - , 5 , = , 5 , - , - , 2 , | , 0 , 0 , |
2 , $ , 0 , | , 1 , - , - , 3 , - , 5 , = , 2 , |
| , 5 , = , 6 , - , - , 1 , | , 0 , | , 0 , 0 , |
1 , | , 0 , $ , 0 , 0 , 0 , | , 0 , 2 , - , - , 2
0 , 2 , - , 4 , - , 2 , - , 3 , - , - , - , 1 , 0
```

h) <u>Test case 8 - input-13(13x13):</u>

- Sử dụng backtracking:

```
Loaded grid:
[1, 0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 2, 0, 0, 2]
[0, 0, 0, 2, 0, 0, 3, 0, 4, 0, 2, 0, 3, 0, 0, 3, 0]
[3, 0, 4, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 1, 0, 1, 0, 0, 4, 0, 6, 0, 0, 3, 0, 1, 0, 0, 0]
[1, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 0]
[0, 3, 0, 0, 1, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2]
[2, 0, 2, 0, 0, 4, 0, 0, 5, 0, 3, 0, 4, 0, 0, 6, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 2, 0, 3, 0, 3, 0, 0, 4, 0, 0, 3, 0, 2, 0, 4, 0]
[2, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 2, 0, 2, 0, 0, 3, 0, 7, 0, 0, 6, 0, 0, 1, 0, 2]
[2, 0, 3, 0, 0, 3, 0, 1, 0, 0, 0, 0, 1, 0, 0, 3, 0]
[0, 5, 0, 0, 5, 0, 6, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 6, 0, 3]
[0, 3, 0, 0, 3, 0, 4, 0, 3, 0, 2, 0, 0, 0, 0, 0, 0]
[2, 0, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 0, 4, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[√] Backtracking chạy trong 386.1872 giây
```

```
Solution found:
1 , 0 , 2 , - , 2 , - , - , 2 , - , 2 , - , 2 , - , 2 , - , - , 2
| , 0 , | , 2 , - , - , 3 , - , 4 , - , 2 , - , 3 , = , = , 3 , |
3 , = , 4 , | , 0 , 2 , | , 0 , $ , 0 , 0 , 0 , 0 , 0 , 0 , | , 2
0 , 1 , | , 1 , 0 , $ , 4 , = , 6 , - , - , 3 , - , 1 , 0 , | , |
1 , | , 2 , - , - , 4 , | , 0 , | , 0 , 0 , | , 2 , - , - , 4 , |
| , 3 , - , - , 1 , | , 1 , 0 , | , 1 , - , 2 , | , 0 , 0 , $ , 2
2 , | , 2 , - , - , 4 , - , - , 5 , = , 3 , - , 4 , = , = , 6 , |
| , | , | , 0 , 0 , | , 1 , 0 , | , 0 , 0 , 0 , 0 , 0 , 0 , $ , 2
| , 2 , | , 3 , - , 3 , | , 0 , 4 , - , - , 3 , - , 2 , - , 4 , |
2 , | , 3 , $ , 0 , | , | , 0 , $ , 0 , 0 , | , 0 , 0 , 0 , | , |
| , 2 , $ , 2 , 0 , | , 3 , = , 7 , = , = , 6 , - , - , 1 , | , 2
2 , | , 3 , - , - , 3 , - , 1 , | , 0 , 0 , $ , 1 , - , - , 3 , |
| , 5 , = , = , 5 , = , 6 , = , 5 , - , - , 6 , = , = , 4 , | , 3
| , $ , 0 , 0 , | , 0 , $ , 0 , | , 0 , 1 , | , 0 , 0 , $ , 1 , $
2 , $ , 0 , 0 , | , 0 , $ , 0 , | , 0 , | , 2 , - , - , 6 , - , 3
| , 3 , - , - , 3 , - , 4 , - , 3 , - , 2 , 0 , 0 , 0 , $ , 0 , 0
2 , - , - , - , 2 , - , 2 , - , 2 , - , 2 , - , 2 , - , - , 4 , - , 1
```

- Sử dụng pysat:

```
Loaded grid:
[1, 0, 2, 0, 2, 0, 0, 2, 0, 2, 0, 2, 0, 2, 0, 0, 2]
[0, 0, 0, 2, 0, 0, 3, 0, 4, 0, 2, 0, 3, 0, 0, 3, 0]
[3, 0, 4, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 1, 0, 1, 0, 0, 4, 0, 6, 0, 0, 3, 0, 1, 0, 0, 0]
[1, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 0]
[0, 3, 0, 0, 1, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 2]
[2, 0, 2, 0, 0, 4, 0, 0, 5, 0, 3, 0, 4, 0, 0, 6, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
[0, 2, 0, 3, 0, 3, 0, 0, 4, 0, 0, 3, 0, 2, 0, 4, 0]
[2, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 2, 0, 2, 0, 0, 3, 0, 7, 0, 0, 6, 0, 0, 1, 0, 2]
[2, 0, 3, 0, 0, 3, 0, 1, 0, 0, 0, 0, 1, 0, 0, 3, 0]
[0, 5, 0, 0, 5, 0, 6, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 6, 0, 3]
[0, 3, 0, 0, 3, 0, 4, 0, 3, 0, 2, 0, 0, 0, 0, 0, 0]
[2, 0, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 0, 4, 0, 1]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[✓] Tìm được lời giải lần 1, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 2, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 3, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 4, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 5, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 6, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 7, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 8, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 9, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 10, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 11, kiểm tra liên thông...
[x] Lời giải không liên thông, thử lại...
[✓] Tìm được lời giải lần 12, kiểm tra liên thông...
[✓] Lời giải liên thông hợp lệ.
[✓] PySAT chạy trong 0.0093 giây

Solution found:
1 , 0 , 2 , – , 2 , – , – , 2 , – , 2 , – , 2 , – , 2 , – , – , 2
| , 0 , | , 2 , – , – , 3 , – , 4 , – , 2 , – , 3 , = , = , 3 , |
3 , = , 4 , | , 0 , 2 , | , 0 , $ , 0 , 0 , 0 , 0 , 0 , 0 , | , 2
0 , 1 , | , 1 , 0 , $ , 4 , = , 6 , – , – , 3 , – , 1 , 0 , | , |
1 , | , 2 , – , – , 4 , | , 0 , | , 0 , 0 , | , 2 , – , – , 4 , |
| , 3 , – , – , 1 , | , 1 , 0 , | , 1 , – , 2 , | , 0 , 0 , $ , 2
2 , | , 2 , – , – , 4 , – , – , 5 , = , 3 , – , 4 , = , = , 6 , |
| , | , | , 0 , 0 , | , 1 , 0 , | , 0 , 0 , 0 , 0 , 0 , 0 , $ , 2
| , 2 , | , 3 , – , 3 , | , 0 , 4 , – , – , 3 , – , 2 , – , 4 , |
2 , | , 3 , $ , 0 , | , | , 0 , $ , 0 , 0 , | , 0 , 0 , 0 , | , |
| , 2 , $ , 2 , 0 , | , 3 , = , 7 , = , = , 6 , – , – , 1 , | , 2
2 , | , 3 , – , – , 3 , – , 1 , | , 0 , 0 , $ , 1 , – , – , 3 , |
| , 5 , = , = , 5 , = , 6 , = , 5 , – , – , 6 , = , = , 4 , | , 3
| , $ , 0 , 0 , | , 0 , $ , 0 , | , 0 , 1 , | , 0 , 0 , $ , 1 , $
2 , $ , 0 , 0 , | , 0 , $ , 0 , | , 0 , | , 2 , – , – , 6 , – , 3
| , 3 , – , – , 3 , – , 4 , – , 3 , – , 2 , 0 , 0 , 0 , $ , 0 , 0
2 , – , – , 2 , – , 2 , – , 2 , – , 2 , – , 2 , – , – , 4 , – , 1
```

i) Test case 9 - input-14(17x17):

- Sử dụng module pysat:

```
Loaded grid:
[1, 0, 2, 0, 0, 3, 0, 3, 0, 3, 0, 2, 0, 2, 0, 1, 0]
[0, 1, 0, 0, 2, 0, 0, 0, 1, 0, 3, 0, 5, 0, 5, 0, 4]
[2, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3]
[2, 0, 5, 0, 3, 0, 3, 0, 5, 0, 6, 0, 8, 0, 4, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[2, 0, 7, 0, 0, 4, 0, 1, 0, 1, 0, 0, 6, 0, 5, 0, 3]
[0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0]
[3, 0, 8, 0, 0, 5, 0, 6, 0, 0, 3, 0, 6, 0, 8, 0, 4]
[0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 2, 0, 1, 0, 0, 0]
[3, 0, 6, 0, 4, 0, 0, 3, 0, 1, 0, 0, 3, 0, 8, 0, 4]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 0]
[0, 0, 6, 0, 8, 0, 4, 0, 6, 0, 3, 0, 2, 0, 4, 0, 3]
[3, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 3]
[2, 0, 4, 0, 4, 0, 2, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0]
[0, 1, 0, 2, 0, 2, 0, 3, 0, 3, 0, 3, 0, 2, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[✓] Tìm được lời giải lần 1, kiểm tra liên thông...
[✓] Lời giải liên thông hợp lệ.
[✓] PySAT chạy trong 0.0158 giây
```

```
Solution found:
1 , – , 2 , – , – , 3 , – , 3 , – , 3 , – , 2 , – , 2 , – , 1 , 0
0 , 1 , – , – , 2 , | , 0 , | , 1 , | , 3 , – , 5 , = , 5 , = , 4
2 , – , – , 1 , | , | , 1 , | , | , 1 , $ , 0 , $ , 0 , | , 0 , $
| , 0 , 0 , 0 , | , 2 , | , 1 , | , 0 , $ , 0 , $ , 0 , | , 0 , 3
2 , – , 5 , = , 3 , | , 3 , – , 5 , = , 6 , = , 8 , = , 4 , 0 , |
0 , 0 , $ , 0 , 0 , | , | , 0 , | , 0 , 0 , 0 , $ , 0 , | , 0 , |
2 , – , 7 , = , = , 4 , | , 1 , | , 1 , – , – , 6 , – , 5 , – , 3
| , 0 , $ , 0 , 0 , | , 1 , | , 2 , – , – , 1 , $ , 0 , $ , 0 , |
3 , = , 8 , = , = , 5 , = , 6 , – , – , 3 , = , 6 , = , 8 , = , 4
0 , 0 , $ , 0 , 0 , 0 , 0 , $ , 3 , – , – , 2 , – , 1 , $ , 0 , |
3 , – , 6 , – , 4 , – , – , 3 , $ , 1 , – , – , 3 , = , 8 , = , 4
$ , 0 , $ , 0 , $ , 0 , 0 , 0 , $ , 0 , 0 , 3 , – , 1 , $ , 0 , |
$ , 0 , 6 , = , 8 , = , 4 , – , 6 , = , 3 , $ , 2 , – , 4 , – , 3
3 , 0 , $ , 0 , $ , 0 , | , 0 , | , 1 , | , 3 , | , 0 , 0 , 0 , |
| , 0 , $ , 0 , $ , 0 , | , 1 , | , | , 1 , | , | , 1 , – , – , 3
2 , – , 4 , – , 4 , – , 2 , | , 1 , | , 0 , | , 2 , – , – , 1 , |
0 , 1 , – , 2 , – , 2 , – , 3 , – , 3 , – , 3 , – , 2 , – , – , 2
```

- Sử dụng Backtracking:

```
Loaded grid:
[1, 0, 2, 0, 0, 3, 0, 3, 0, 3, 0, 2, 0, 2, 0, 1, 0]
[0, 1, 0, 0, 2, 0, 0, 0, 1, 0, 3, 0, 5, 0, 5, 0, 4]
[2, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3]
[2, 0, 5, 0, 3, 0, 3, 0, 5, 0, 6, 0, 8, 0, 4, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[2, 0, 7, 0, 0, 4, 0, 1, 0, 1, 0, 0, 6, 0, 5, 0, 3]
[0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0]
[3, 0, 8, 0, 0, 5, 0, 6, 0, 0, 3, 0, 6, 0, 8, 0, 4]
[0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 2, 0, 1, 0, 0, 0]
[3, 0, 6, 0, 4, 0, 0, 3, 0, 1, 0, 0, 3, 0, 8, 0, 4]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 0]
[0, 0, 6, 0, 8, 0, 4, 0, 6, 0, 3, 0, 2, 0, 4, 0, 3]
[3, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 3]
[2, 0, 4, 0, 4, 0, 2, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0]
[0, 1, 0, 2, 0, 2, 0, 3, 0, 3, 0, 3, 0, 2, 0, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[✓] Backtracking chạy trong 327.1323 giây
```

```
Solution found:
1 , - , 2 , - , - , 3 , - , 3 , - , 3 , - , 2 , - , 2 , - , 1 , 0
0 , 1 , - , - , 2 , | , 0 , | , 1 , | , 3 , - , 5 , = , 5 , = , 4
2 , - , - , 1 , | , | , 1 , | , | , 1 , $ , 0 , $ , 0 , | , 0 , $
| , 0 , 0 , 0 , | , 2 , | , 1 , | , 0 , $ , 0 , $ , 0 , | , 0 , 3
2 , - , 5 , = , 3 , | , 3 , - , 5 , = , 6 , = , 8 , = , 4 , 0 , |
0 , 0 , $ , 0 , 0 , | , | , 0 , | , 0 , 0 , 0 , $ , 0 , | , 0 , |
2 , - , 7 , = , = , 4 , | , 1 , | , 1 , - , - , 6 , - , 5 , - , 3
| , 0 , $ , 0 , 0 , | , 1 , | , 2 , - , - , 1 , $ , 0 , $ , 0 , |
3 , = , 8 , = , = , 5 , = , 6 , - , - , 3 , = , 6 , = , 8 , = , 4
0 , 0 , $ , 0 , 0 , 0 , 0 , $ , 3 , - , - , 2 , - , 1 , $ , 0 , |
3 , - , 6 , - , 4 , - , - , 3 , $ , 1 , - , - , 3 , = , 8 , = , 4
$ , 0 , $ , 0 , $ , 0 , 0 , 0 , $ , 0 , 0 , 3 , - , 1 , $ , 0 , |
$ , 0 , 6 , = , 8 , = , 4 , - , 6 , = , 3 , $ , 2 , - , 4 , - , 3
3 , 0 , $ , 0 , $ , 0 , | , 0 , | , 1 , | , 3 , | , 0 , 0 , 0 , |
| , 0 , $ , 0 , $ , 0 , | , 1 , | , | , 1 , | , | , 1 , - , - , 3
2 , - , 4 , - , 4 , - , 2 , | , 1 , | , 0 , | , 2 , - , - , 1 , |
0 , 1 , - , 2 , - , 2 , - , 3 , - , 3 , - , 3 , - , 2 , - , - , 2
```

j) Test case 10 - input-16(20x20):

- Sử dụng module pysat:

```
Loaded grid:
[3, 0, 3, 0, 0, 3, 0, 2, 0, 2, 0, 0, 3, 0, 3, 0, 0, 2, 0, 0]
[0, 4, 0, 0, 3, 0, 3, 0, 1, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0, 1]
[3, 0, 0, 3, 0, 1, 0, 1, 0, 5, 0, 0, 4, 0, 0, 1, 0, 0, 0, 0]
[0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 5]
[3, 0, 0, 6, 0, 0, 5, 0, 0, 3, 0, 2, 0, 2, 0, 0, 4, 0, 0, 0]
[0, 0, 1, 0, 1, 0, 0, 0, 3, 0, 3, 0, 3, 0, 0, 2, 0, 0, 0, 0]
[0, 2, 0, 3, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 4, 0, 6]
[3, 0, 2, 0, 5, 0, 7, 0, 8, 0, 0, 5, 0, 2, 0, 0, 3, 0, 0, 0]
[0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 4, 0, 4]
[2, 0, 3, 0, 5, 0, 4, 0, 0, 0, 1, 0, 0, 0, 3, 0, 4, 0, 1, 0]
[0, 0, 0, 3, 0, 2, 0, 0, 4, 0, 0, 4, 0, 5, 0, 3, 0, 5, 0, 3]
[4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0]
[0, 0, 0, 6, 0, 0, 4, 0, 6, 0, 6, 0, 0, 5, 0, 0, 4, 0, 0, 3]
[1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 3]
[0, 0, 0, 3, 0, 0, 5, 0, 7, 0, 4, 0, 0, 3, 0, 5, 0, 1, 0, 0]
[2, 0, 5, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0, 1, 0, 0, 0, 0, 3]
[3, 0, 6, 0, 3, 0, 3, 0, 4, 0, 0, 2, 0, 2, 0, 4, 0, 0, 2, 0]
[0, 1, 0, 3, 0, 3, 0, 3, 0, 0, 3, 0, 3, 0, 3, 0, 0, 2, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 1
[✓] Tìm được lời giải lần 1, kiểm tra liên thông...
[✓] Lời giải liên thông hợp lệ.
[✓] PySAT chạy trong 0.0102 giây

Solution found:
3 , - , 3 , = , = , 3 , - , 2 , - , 2 , 0 , 0 , 3 , = , 3 , - , - , 2 , 0 , 0
$ , 4 , = , = , 3 , - , 3 , - , 1 , | , 0 , 0 , | , 1 , - , - , 3 , | , 0 , 1
3 , $ , 0 , 3 , - , 1 , | , 1 , - , 5 , = , = , 4 , - , - , 1 , $ , | , 0 , |
| , 3 , 0 , $ , 0 , 0 , | , 0 , 0 , | , 0 , 0 , 0 , 0 , 0 , 0 , $ , 4 , = , 5
3 , | , 0 , 6 , = , = , 5 , - , - , 3 , - , 2 , - , 2 , - , - , 4 , | , 0 , $
$ , | , 1 , $ , 1 , 0 , | , 0 , 3 , - , 3 , = , 3 , - , - , 2 , | , | , 0 , $
$ , 2 , | , 3 , | , 0 , | , 0 , $ , 1 , - , - , - , - , 2 , | , | , 4 , = , 6
3 , | , 2 , | , 5 , = , 7 , = , 8 , = , = , 5 , - , 2 , | , | , 3 , | , 0 , $
| , 1 , | , 1 , $ , 0 , $ , 0 , $ , 0 , 0 , $ , 0 , | , | , 1 , $ , 4 , - , 4
2 , 0 , 3 , - , 5 , = , 4 , 0 , $ , 0 , 1 , $ , 0 , | , 3 , = , 4 , $ , 1 , |
| , 0 , | , 3 , - , 2 , - , - , 4 , 0 , | , 4 , = , 5 , - , 3 , = , 5 , | , 3
4 , = , 4 , $ , 0 , 0 , 0 , 0 , | , 0 , | , 0 , 0 , | , 0 , 0 , 0 , | , 3 , $
| , 0 , | , 6 , = , = , 4 , - , 6 , = , 6 , - , - , 5 , = , = , 4 , | , $ , 3
1 , 0 , 2 , $ , 0 , 0 , | , 0 , $ , 0 , $ , 0 , 0 , | , 0 , 1 , $ , | , 3 , |
0 , 0 , | , $ , 0 , 0 , | , 0 , $ , 0 , $ , 0 , 1 , | , 0 , | , 3 , | , | , 3
0 , 0 , | , 3 , - , - , 5 , = , 7 , - , 4 , 0 , | , 3 , = , 5 , | , 1 , | , $
2 , - , 5 , - , - , 1 , | , 0 , $ , 0 , | , 0 , | , 0 , 0 , $ , 2 , - , 3 , $
| , 0 , $ , 0 , 0 , 0 , | , 0 , $ , 0 , 2 , - , 3 , - , 1 , $ , 0 , 0 , | , 3
3 , = , 6 , = , 3 , - , 3 , - , 4 , - , - , 2 , - , 2 , - , 4 , - , - , 2 , |
0 , 1 , - , 3 , = , 3 , - , 3 , = , = , 3 , - , 3 , = , 3 , - , - , 2 , - , 2
```

- Sử dụng Backtracking:

```
Loaded grid:
[3, 0, 3, 0, 0, 3, 0, 2, 0, 2, 0, 0, 3, 0, 3, 0, 0, 2, 0, 0]
[0, 4, 0, 0, 3, 0, 3, 0, 1, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0, 1]
[3, 0, 0, 3, 0, 1, 0, 1, 0, 5, 0, 0, 4, 0, 0, 1, 0, 0, 0, 0]
[0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 5]
[3, 0, 0, 6, 0, 0, 5, 0, 0, 3, 0, 2, 0, 2, 0, 0, 4, 0, 0, 0]
[0, 0, 1, 0, 1, 0, 0, 0, 3, 0, 3, 0, 3, 0, 0, 2, 0, 0, 0, 0]
[0, 2, 0, 3, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 4, 0, 6]
[3, 0, 2, 0, 5, 0, 7, 0, 8, 0, 0, 5, 0, 2, 0, 0, 3, 0, 0, 0]
[0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 4, 0, 4]
[2, 0, 3, 0, 5, 0, 4, 0, 0, 0, 1, 0, 0, 0, 3, 0, 4, 0, 1, 0]
[0, 0, 0, 3, 0, 2, 0, 0, 4, 0, 0, 4, 0, 5, 0, 3, 0, 5, 0, 3]
[4, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0]
[0, 0, 0, 6, 0, 0, 4, 0, 6, 0, 6, 0, 0, 5, 0, 0, 4, 0, 0, 3]
[1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 3]
[0, 0, 0, 3, 0, 0, 5, 0, 7, 0, 4, 0, 0, 3, 0, 5, 0, 1, 0, 0]
[2, 0, 5, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 0, 1, 0, 0, 0, 0, 3]
[3, 0, 6, 0, 3, 0, 3, 0, 4, 0, 0, 2, 0, 2, 0, 4, 0, 0, 2, 0]
[0, 1, 0, 3, 0, 3, 0, 3, 0, 0, 3, 0, 3, 0, 3, 0, 0, 2, 0, 2]

Chọn thuật toán giải:
1. PySAT
2. A*
3. Brute-force
4. Backtracking
Nhập số (1 , 2, 3 hoặc 4): 4
[+] Đang giải bằng backtracking (có cắt tỉa + kiểm tra liên thông)...
[✓] Backtracking chạy trong 585.5575 giây

Solution found:
3 , - , 3 , = , = , 3 , - , 2 , - , 2 , 0 , 0 , 3 , = , 3 , - , - , 2 , 0 , 0
$ , 4 , = , = , 3 , - , 3 , - , 1 , | , 0 , 0 , | , 1 , - , - , 3 , | , 0 , 1
3 , $ , 0 , 3 , - , 1 , | , 1 , - , 5 , = , = , 4 , - , - , 1 , $ , | , 0 , |
| , 3 , 0 , $ , 0 , 0 , | , 0 , 0 , | , 0 , 0 , 0 , 0 , 0 , 0 , $ , 4 , = , 5
3 , | , 0 , 6 , = , = , 5 , - , - , 3 , - , 2 , - , 2 , - , - , 4 , | , 0 , $
$ , | , 1 , $ , 1 , 0 , | , 0 , 3 , - , 3 , = , 3 , - , - , 2 , | , | , 0 , $
$ , 2 , | , 3 , | , 0 , | , 0 , $ , 1 , - , - , - , - , 2 , | , | , 4 , = , 6
3 , | , 2 , | , 5 , = , 7 , = , 8 , = , = , 5 , - , 2 , | , | , 3 , | , 0 , $
| , 1 , | , 1 , $ , 0 , $ , 0 , $ , 0 , 0 , $ , 0 , | , | , 1 , $ , 4 , - , 4
2 , 0 , 3 , - , 5 , = , 4 , 0 , $ , 0 , 1 , $ , 0 , | , 3 , = , 4 , $ , 1 , |
| , 0 , | , 3 , - , 2 , - , - , 4 , 0 , | , 4 , = , 5 , - , 3 , = , 5 , | , 3
4 , = , 4 , $ , 0 , 0 , 0 , 0 , | , 0 , | , 0 , 0 , | , 0 , 0 , 0 , | , 3 , $
| , 0 , | , 6 , = , = , 4 , - , 6 , = , 6 , - , - , 5 , = , = , 4 , | , $ , 3
1 , 0 , 2 , $ , 0 , 0 , | , 0 , $ , 0 , $ , 0 , 0 , | , 0 , 1 , $ , | , 3 , |
0 , 0 , | , $ , 0 , 0 , | , 0 , $ , 0 , $ , 0 , 1 , | , 0 , | , 3 , | , | , 3
0 , 0 , | , 3 , - , - , 5 , = , 7 , - , 4 , 0 , | , 3 , = , 5 , | , 1 , | , $
2 , - , 5 , - , - , 1 , | , 0 , $ , 0 , | , 0 , | , 0 , 0 , $ , 2 , - , 3 , $
| , 0 , $ , 0 , 0 , 0 , | , 0 , $ , 0 , 2 , - , 3 , - , 1 , $ , 0 , 0 , | , 3
3 , = , 6 , = , 3 , - , 3 , - , 4 , - , - , 2 , - , 2 , - , 4 , - , - , 2 , |
0 , 1 , - , 3 , = , 3 , - , 3 , = , = , 3 , - , 3 , = , 3 , - , - , 2 , - , 2
```

IV. TỰ ĐÁNH GIÁ:

1. Về nhóm:

- Về khó khăn, ban đầu tụi em khá bối rối, không biết phải giải quyết câu đố theo hướng nào vì thực ra logic là một kiến thức vô cùng mới mẻ đối tụi em. Nhưng sau một thời gian tìm hiểu thì các bạn cũng đã có một nền tảng nhất định và tìm ra hướng đi của đồ án.
- Khó khăn tiếp theo chính là việc thêm các điều kiện bài toán ở dạng logic sao cho phù hợp để tìm ra đúng lời giải. Chúng em đã phải thử nghiệm rất nhiều để tìm ra được đáp án cuối cùng.
- Về lợi ích, chúng em có được nhiều kiến thức hơn, hiểu được cách giải một bài toán logic và quan trọng nhất là rèn dũa kỹ năng làm việc nhóm.
- Nhìn chung thì các bạn đều rất cố gắng và sôi nổi trong quá trình làm việc nhóm. Nhờ vào điều đó mà nhóm em đã hoàn thành tốt dự án này.

2. Về thuật toán:

- Giải bằng module pysat là nhanh nhất và tối ưu nhất.
- Giải bằng brute-force là chậm nhất và có khả năng không ra nổi đối với những bài toán có kích thước lớn và chỉ giải tối đa được 6x6.
- Các thuật toán khác thì có khả năng chạy được với những input có kích thước nhỏ và trung bình, cụ thể là A* giải được tới 9x9 và backtrack sẽ giải được tới 17x17

V. LINK:

- Video demo: **https://www.youtube.com/watch?v=2J_yDjF_Y38**
- Github:**https://github.com/LongLee0/Hcmus-Project-Hashiwokakero.git**

VI. REFERENCE:

- ChatGPT: Phân tích tối ưu logic, khái quát lại cấu trúc chương trình để vá lỗi, cung cấp kiến thức mới.
- Github: https://github.com/erthium/hashiwokakero
- Hỗ trợ tạo các map input: https://www.hashi.info/
- Tham khảo backtracking:
  + https://github.com/lalt22/AIHashi
  + https://github.com/NQAnh-HCMUS/Intro2AI_02-Hashiwokakero

—----------**END**—----------