ITEC

CS-300 Artificial Intelligence

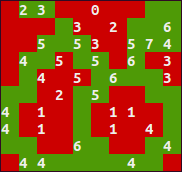
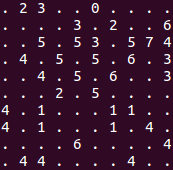
CNF

***Dr. Nguyen Ngoc Thao***

***Msc. Nguyen Hai Dang - Msc. Do Trong Le – Nguyen Quang Thuc***

# Mô tả bài toán Puzzle

Cho một ma trận có kích thước m x n, trong đó mỗi ô sẽ là một con số nguyên không âm hoặc không có giá trị (ô trống). Mỗi ô sẽ có 8 ô liền kề và ở đây bản thân ô đó cũng được tính là liền kề với chính nó. Người chơi sẽ phải tô màu đỏ và xanh vào tất cả các ô trên ma trận, sao cho số ô xanh “liền kề” với một ô đúng bằng con số bên trong ô đó.



Hình 1. Ví dụ về một ma trận input (trái) và đáp án (phải)

Để giải bài toán về Puzzle này, có rất nhiều cách. Trong nội dung môn học này, chúng ta sử dụng mệnh đề logic bậc nhất dạng CNF để giải. Trong đó, đặt vào mỗi ô trên ma trận một biến logic (nhận giá trị True thì ô đó được tô màu xanh và False thì tô màu đỏ), các ràng buộc được viết ra cho mỗi ô có số để thu được một tập hợp các mệnh đề ràng buộc dạng CNF. Sau đó, dùng thư viện [pysat](https://pysathq.github.io/usage.html) để giải, ta tìm được giá trị True/False cho từng biến và suy ra được kết quả.

Ví dụ để ràng buộc đối với một ô chứa số, ta đặt các biến logic để viết mệnh đề.

| . | . | . |  | a | b | c |
| --- | --- | --- | --- | --- | --- | --- |
| . | 2 | . |  | d | e | f |
| . | . | . |  | g | h | i |

# Gợi ý thuật toán

* Duyệt qua từng ô có chứa số để viết các mệnh đề ràng buộc dạng CNF
* Tổng hợp các mệnh đề và lọc bỏ trùng lặp

Cách viết ràng buộc cho mỗi ô chứa số, ví dụ trường hợp số 2

* Đặt 9 biến logic (a, b, c, …, i) ứng với 8 ô xung quanh và ô tại số 2
  + nếu một biến là True thì ô của nó màu xanh
  + nếu một biến là False thì ô của nó màu đỏ

| . | . | . |  | a | b | c |
| --- | --- | --- | --- | --- | --- | --- |
| . | 2 | . |  | d | e | f |
| . | . | . |  | g | h | i |

* Giả sử ta chọn ô ứng với biến a, b là màu xanh thì tương đương với điều này 7 ô còn lại phải là màu đỏ

a ^ b ↔-c ^ -d ^ -e ^ -f ^ -g ^ -h ^ -i

* Biến đổi mệnh đề kéo theo hai chiều trên thành CNF ta được

(-a V -b) V (-c ^ -d ^ -e ^ -f ^ -g ^ -h ^ -i)

and

(c V d V e V f V g V h V i) V (a ^ b)

* Phân phối vào ta được

| -a V -b V -c  -a V -b V -d  -a V -b V -e  -a V -b V -f  -a V -b V -g  -a V -b V -h  -a V -b V -i | and | c V d V e V f V g V h V i V a  c V d V e V f V g V h V i V b |
| --- | --- | --- |

* Tương tự cho các bộ 2 biến còn lại, có **C2n** (n là số ô kề) trường hợp như vậy.
* Ta rút ra một quy luật ở đây: xung quanh ô số 2,
  + ta chọn ra 3 biến bất kỳ x, y, z để viết thành các mệnh đề dạng

(-x V -y V -z) [**CASE 1**]

* + và chọn ra 8 biến bất kỳ q, w, e, r, t, y, u, i để viết các mệnh đề dạng

(q V w V e V r V t V y V u V i) [**CASE 2**]

* Tổng quát lên với mỗi ô mang số K, ta chọn ra **K+1** biến để viết các mệnh đề dạng [**CASE 1**] và chọn ra **số\_ô\_kề - K + 1** để viết các mệnh đề dạng [**CASE 2**]. Đây chính là quy luật dùng để lập trình.

# Dữ liệu lập trình

| input.txt (tab-seperated) | output (terminal) |
| --- | --- |
| 3 3  0 . 1  . . .  2 . 1 |  |

# Yêu cầu

* Dạng bài tập: Cá nhân
* Yêu cầu: Hoàn chỉnh mã nguồn trong file puzzle.py được cung cấp để giải bài toán puzzle bằng CNF và thư viện **pysat**.

# Quy định nộp bài

Nộp bài trên moodle theo đúng cấu trúc tên sau:

* Bài nộp gồm đúng 01 file .zip chứa
  + 01 file mã nguồn **puzzle.py** đã được hoàn thiện và có thể biên dịch (có thể chứa các file modules khác).
  + Ảnh chụp màn hình minh hoạ kết quả chạy của chương trình: **demo.png**
* File .zip được đặt tên theo cú pháp **MSSV\_HoTen.zip** (Vd: *1752001\_NguyenVanA.zip*)

# Lưu ý

* Sai cú pháp nộp bài sẽ mất điểm của homework này.
* Chép code của nhau 0 điểm!
* Nếu các bạn tham khảo nguồn nào, bắt buộc phải ghi nguồn tham khảo vào trong file code dưới dạng chú thích (comment). Những bài không trích dẫn nguồn tham khảo sẽ bị 0 điểm.