BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



**BÁO CÁO ĐỒ ÁN HỌC PHẦN**

**TRÍ TUỆ NHÂN TẠO – CT332**

**Đề tài**

**Áp dụng giải thuật Mô Phỏng Luyện Kim**

**với bài toán Sudoku**

**Sinh viên thực hiện:**

**Phạm Hoàng Phong B2113316**

**Lê Phú Thịnh B2113320**

**Nguyễn Minh Tâm B2113394**

Cần Thơ, 11/2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN 2](#_Toc150893827)

[1. Sudoku là gì? 2](#_Toc150893828)

[2. Luật chơi Sudoku 2](#_Toc150893829)

[3. Cách giải bài toán Sudoku 4](#_Toc150893830)

[CHƯƠNG 2: GIẢI THUẬT MÔ PHỎNG LUYỆN KIM SIMULATED ANNEANLING (SA) 6](#_Toc150893831)

[1. Giới thiệu 6](#_Toc150893832)

[2. Ý tưởng 6](#_Toc150893833)

[CHƯƠNG 3 ÁP DỤNG MÔ PHỎNG LUYỆN KIM ĐỂ GIẢI BÀI TOÁN SUDOKU 10](#_Toc150893834)

[1. Mục tiêu 10](#_Toc150893835)

[2. Phân tích bài toán 10](#_Toc150893836)

[3. Cài đặt giải thuật 12](#_Toc150893837)

[PHẦN KẾT LUẬN 19](#_Toc150893838)

[1. Kết quả đạt được 19](#_Toc150893839)

[2. Hướng phát triển 19](#_Toc150893840)

[3. Hạn chế 19](#_Toc150893841)

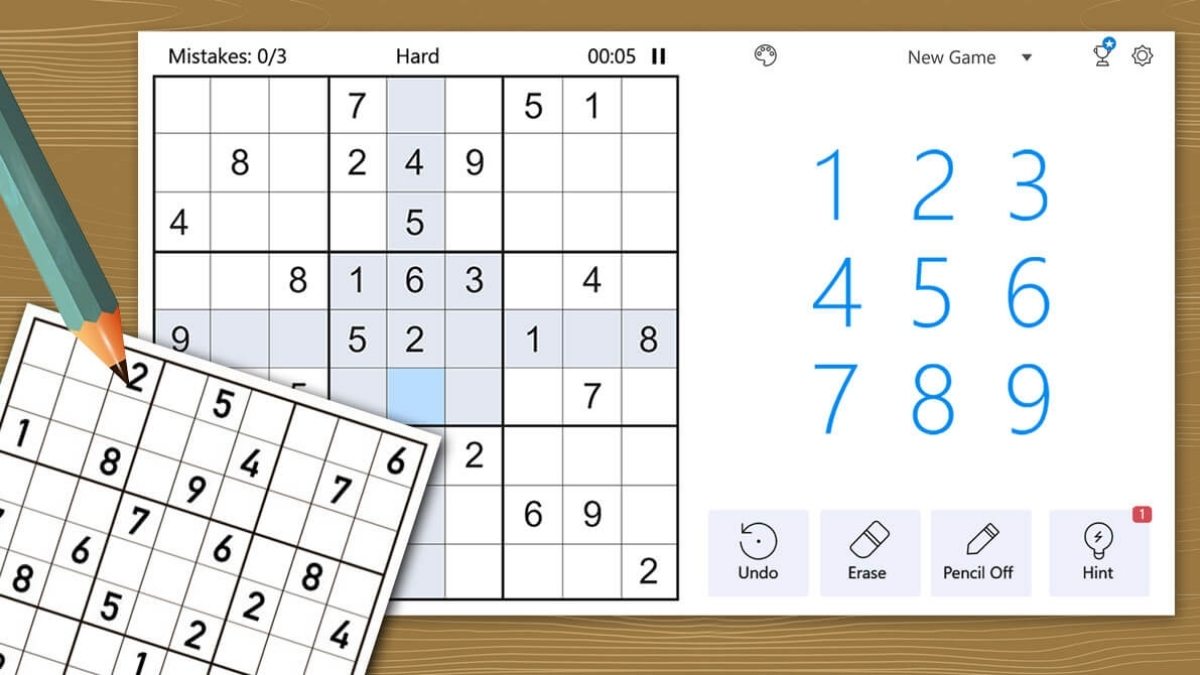
[TÀI LIỆU THAM KHẢO 20](#_Toc150893842)

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## Sudoku là gì?

**Sudoku**, ban đầu có tên là **Number Place**, có nguồn gốc từ nước Mỹ. **Sudoku** là một trò chơi câu đố sắp xếp chữ số dựa trên logic theo tổ hợp.

Mục tiêu của trò chơi là điền các chữ số vào khối 9×9 sao cho mỗi cột, mỗi hàng, và mỗi khối con 3×3 cấu tạo nên khối chính đều chứa tất cả các chữ số từ 1 tới 9. Câu đố đã được hoàn thành trước một phần, người chơi phải giải tiếp bằng việc điền số. Mỗi câu đố được thiết lập tốt có một cách làm duy nhất.



Hình Trò chơi Sudoku

## Luật chơi Sudoku

Luật chơi vô cùng đơn giản, tất cả những gì người chơi cần làm là điền những chữ số từ 1 tới 9 vào những vị trí phù hợp, tuân theo những quy luật đã được định sẵn.

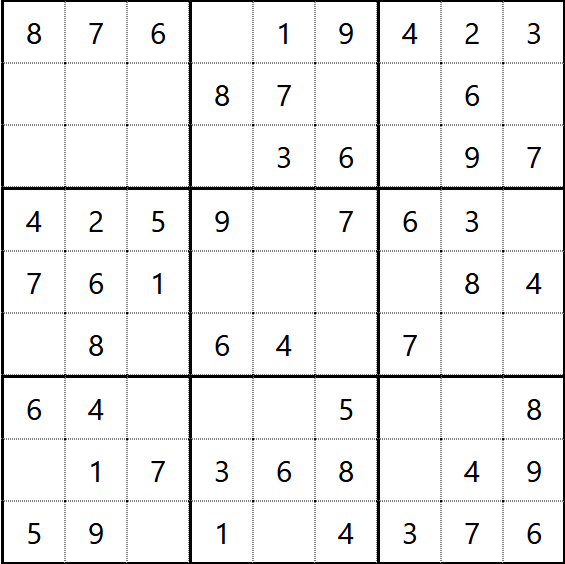
Mỗi một ô vuông nhỏ trong lưới sẽ thuộc một hàng, một cột và một khối 3×3 tương ứng.

Chữ số được điền vào 1 ô trống sẽ phải đảm bảo tuân theo 3 điều kiện:

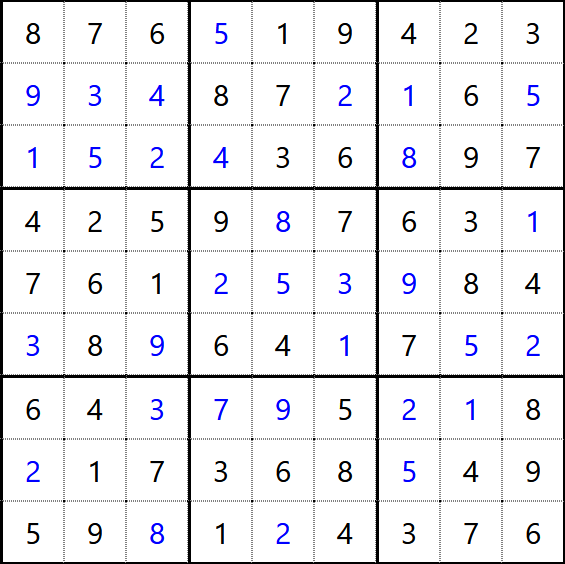
* Là chữ số duy nhất trong **hàng**.
* Là chữ số duy nhất trong **cột**.
* Là chữ số duy nhất trong **khối**.

Một lời giải hoàn chỉnh sẽ gồm **3** yếu tố:

* **Mỗi hàng ngang** có đủ 9 chữ số (từ 1 tới 9), không cần theo đúng thứ tự.
* **Mỗi hàng dọc** có đủ 9 chữ số.
* **Trong mỗi khối vuông 3×3 (tổng cộng 9 khối)** cũng có mặt đủ 9 chữ số khác nhau.



Hình Khối Sudoku chưa giải



Hình Khối Sudoku hoàn thiện

Ngoài loại khối 9×9, Sudoku còn có những biến thể khác như:

* Khối 4×4 chia thành các khối con 2×2.
* Kích thước 5×5 ô chia vùng theo Pentomino (được phát hành với tên gọi Logi-5).
* Kích thước 7×7 ô chia vùng theo Heptomino.
* Khối 100×100 (là biến thể Sudoku có kích thước lớn nhất hiện nay).
* …

## Cách giải bài toán Sudoku

Giải quyết một bài toán Sudoku đòi hỏi sự kỹ năng trong việc áp dụng logic và phân tích để điền các số vào lưới 9×9 sao cho đảm bảo các quy tắc đã được nêu trên.

Để tiến hành giải bài toán Sudoku, chúng ta có thể áp dụng một số kỹ thuật:

* Điền vào các ô rõ ràng (các ô chỉ có 1 lựa chọn duy nhất dựa vào các số đã được điền sẵn vào đề bài).
* Quan sát và loại trừ dựa theo số trên hàng ngang, hàng dọc và trong 1 khối.
* Kỹ thuật Nhóm (Naked Pairs, Naked Triples, Hidden Pairs, Hidden Triples).
* …

Người chơi có thể áp dụng và kết hợp các kỹ thuật trên để giải được Sudoku một cách tối ưu hơn, có thể sẽ cần thử nghiệm và lặp lại các bước để điều chỉnh lựa chọn của mình.

Trong **lập trình**, Sudoku là 1 dạng bài toán phổ biến và chúng ta có thể sử dụng nhiều loại giải thuật khác nhau để tìm ra lời giải, một số giải thuật được sử dụng để giải Sudoku như:

* Giải thuật **Quay lui** **(Backtracking):** Quay lui là một giải thuật dựa trên đệ quy. Ý tưởng của quay lui là tìm lời giải từng bước, mỗi bước chọn một trong số các lựa chọn khả dĩ và đệ quy.
* Giải thuật **Di truyền** **(Genetic Algorithm):** Là một giải thuật tìm kiếm cục bộ, trong đó bạn cố gắng cải thiện lời giải hiện tại bằng cách thay đổi một số yếu tố nhỏ.
* Giải thuật **Mô phỏng luyện kim** **(Simulated Annealing):** Là giải thuật tìm kiếm xác suất, dựa trên quá trình luyện kim trong vật lý, để tìm ra giá trị tối ưu toàn cục cho một hàm mục tiêu.

Với các giải thuật khác nhau sẽ có ưu điểm và nhược điểm riêng đối với từng bài toán và trong phạm vi của bài báo cáo này nhóm chúng em sẽ đi sâu vào sử dụng giải thuật **Mô phỏng luyện kim** để giải bài toán **Sudoku**.

# CHƯƠNG 2: GIẢI THUẬT MÔ PHỎNG LUYỆN KIM SIMULATED ANNEANLING (SA)

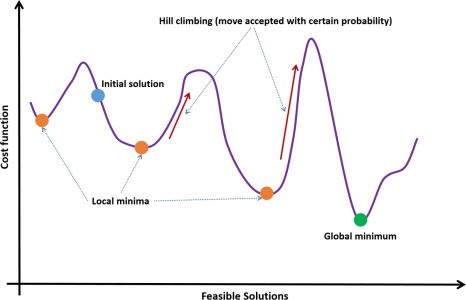
## Giới thiệu

Đây là biến thể của giải thuật Metropolis – Hastings, được công bố bởi Metropolis và cộng sự vào năm 1953.

**Simulated Annealing (SA)** lấy cảm hứng từ sự tương đồng giữa quá trình luyện kim vật lý của chất rắn và giải quyết các bài toán tối ưu hoá tổ hợp lớn.

**Simulated Annealing (SA)** là gì?

* SA là giải thuật được dùng để giải quyết bài toán tối ưu hoá tổ hợp.
* SA là một thuật toán dựa trên xác suất.
* SA cho phép bước di chuyển không tốt để thoát khỏi vùng tối ưu cục bộ.
* SA dễ để cài đặt và triển khai.
* SA được cải tiến từ thuật toán leo đồi (Hill Climbing Algorithm).



Hình SA chấp nhận bước đi xấu để thoát local optima.

## Ý tưởng

* **Kỹ thuật luyện kim trong vật lý**

Kỹ thuật luyện kim là quá trình nhiệt để đạt được trạng thái năng lượng thấp của chất rắn trong bể nhiệt.

Quá trình gồm 2 bước:

* + **Tăng nhiệt độ:** tăng nhiệt độ trong bể nhiệt đến giá trị tối đa để có thể nung chảy được chất rắn.
  + **Làm mát:** giảm nhiệt độ một cách cẩn thận cho đến khi chất rắn đạt đến trạng thái cơ bản (ground state).

**Trạng thái cơ bản** của chất rắn được hình thành khi giá trị tối đa của nhiệt độ là đủ lớn và quá trình làm mát đủ chậm.

**→ Chất rắn bền** được hình thành từ sự làm mát cẩn thận và chậm rãi.

**Metastable state (trạng thái giả bền)** được hình thành nếu nhiệt độ ban đầu không đủ lớn và quá trình làm lạnh diễn ra nhanh.

Quá trình dẫn đến hình thành Metastable state được gọi là **“Tôi” (Quenching).**

Nếu quá trình hạ nhiệt diễn ra đủ chậm, chất rắn sẽ đến trạng thái **“Cân bằng nhiệt” (Thermal Equilibrium)** ở mỗi mức nhiệt độ khác nhau.

* **Giải thuật mô phỏng luyện kim trong lập trình**

|  |  |
| --- | --- |
| **Physical System** | **Optimization Problem** |
| System state | Solution |
| Molecular positions | Decision Variables |
| Energy | Objective Function |
| Minimizing Energy | Minimizing cost |
| Ground state | Global optimal solution |
| Metastable state | Local optimum |
| Quenching | Local search |
| Temperature | Control parameter T |
| Real annealing | Simulated annealing |

Bảng so sánh kĩ thuật luyện kim vật lý và giải thuật Mô phỏng luyện kim

Giải thuật **Mô phỏng luyện kim:**

**function** SIMULATED-ANNEALING (*problem*, *schedule*) **returns** a solution state

inputs: *problem*, a problem

*schedule*, a mapping from time to “temperature”

*current* 🡨 MAKE-NODE (*problem*.INITIAL-STATE)

**for** t = 1 **to** ∞ **do**

*T* 🡨 schedule(t)

**if** *T* = 0 **then** **return** *current*

*next* 🡨 a randomly selected successor of *current*

△*E* 🡨 *next*.VALUE – *current*.VALUE

**if** △*E* > 0 **then** *current* 🡨 *next*

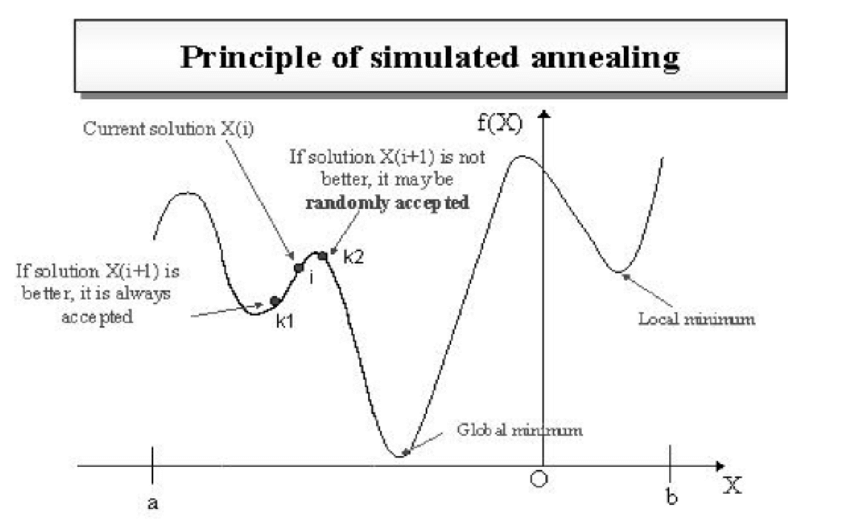
**else** *current* 🡨 *next* only with probability

Trên cơ sở lý thuyết, giải thuật tuân theo một điều kiện cơ bản:

* Nếu sự thay đổi làm giảm giá trị (năng lượng - E) của hàm mục tiêu thì nó được chấp nhận.
* Nếu sự thay đổi làm tăng giá trị của hàm mực tiêu, nó vẫn được chấp nhận nhưng với một xác suất tính bằng công thức
* Ở nhiệt độ (T) cao, P sẽ gần giá trị 1, do đó đa phần các bước di chuyển đều được chấp nhận và giải thuật sẽ trở thành bước đi ngẫu nhiên trong không gian trạng thái.
* Ở nhiệt độ (T) thấp, P sẽ dần về 0, do đó đa phần các bước di chuyển tăng năng lượng (E) đều bị từ chối.

Quá trình thực hiện giải thuật:

* Ở các trạng thái ban đầu, SA tiến hành qua nhiều vòng lặp. Qua mỗi vòng lặp, một trạng thái lân cận ngẫu nhiên được sinh ra.
* Các bước mà hàm chi phí có sự cải thiện sẽ luôn được chấp nhận.
* Ngược lại, trạng thái lân cận sẽ được chọn với một xác suất dựa vào nhiệt độ hiện tại và sự chênh lệch của hàm mục tiêu.
* đại diện cho sự khác biệt của giá trị mục tiêu giữa trạng thái hiện tại và trạng thái lân cận mới được tạo ra.
* Khi thuật toán tiến triển, xác suất mà các bước di chuyển được chấp nhận sẽ giảm xuống.



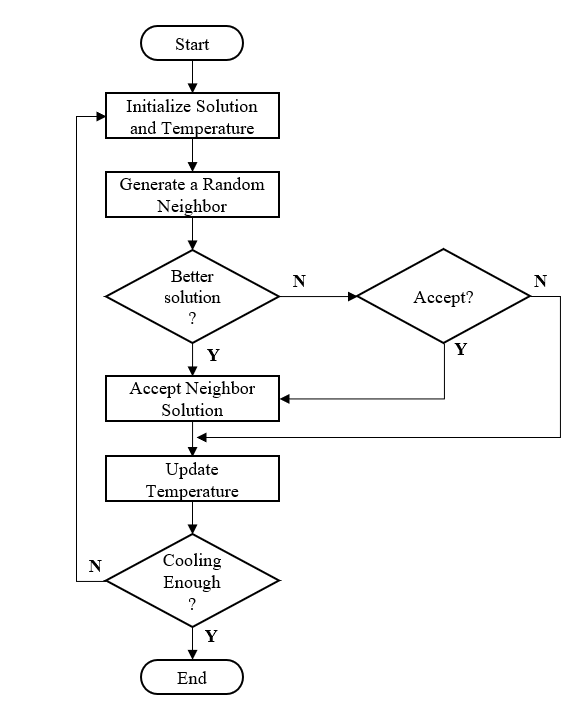
Hình Quá trình tìm kiếm của SA

**Hàm tính xác suất chấp nhận (Acceptance Probability Function),** nhìn chunglà phân phối Boltzmann:

Sử dụng 1 tham số kiểm soát gọi là nhiệt độ (T) và mức chênh lệch năng lượng (hàm mục tiêu) để xác định xác suất chấp nhận một bước đi không có cải thiện.

Tại một mức nhiệt độ nhất định, rất nhiều trạng thái được khám phá.

Khi đã đạt **Trạng thái cân bằng (Equilibrium State),** nhiệt độ sẽ giảm dần xuống dựa theo tiến trình làm mát sao cho chỉ có một số ít trạng thái không có sự cải thiện sẽ được chấp nhận ở cuối quá trình tìm kiếm.



Hình Lưu đồ giải thuật SA

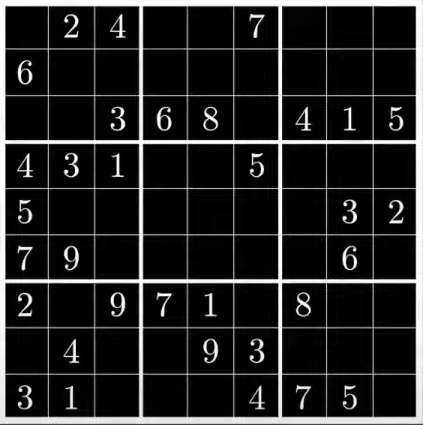
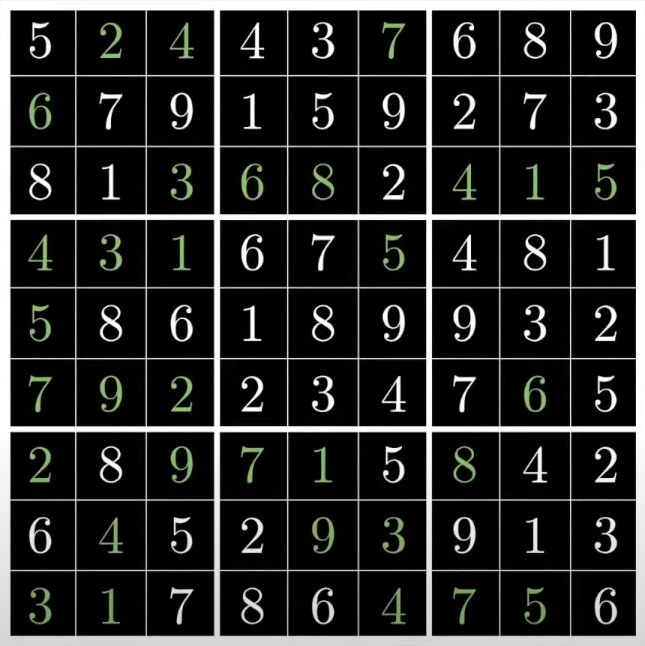
# CHƯƠNG 3 ÁP DỤNG MÔ PHỎNG LUYỆN KIM ĐỂ GIẢI BÀI TOÁN SUDOKU

## Mục tiêu

* Biểu diễn được trạng thái của bài toán và áp dụng giải thuật **Mô phỏng luyện kim (SA)** để giải bài toán.
* Sử dụng Ngôn ngữ C.

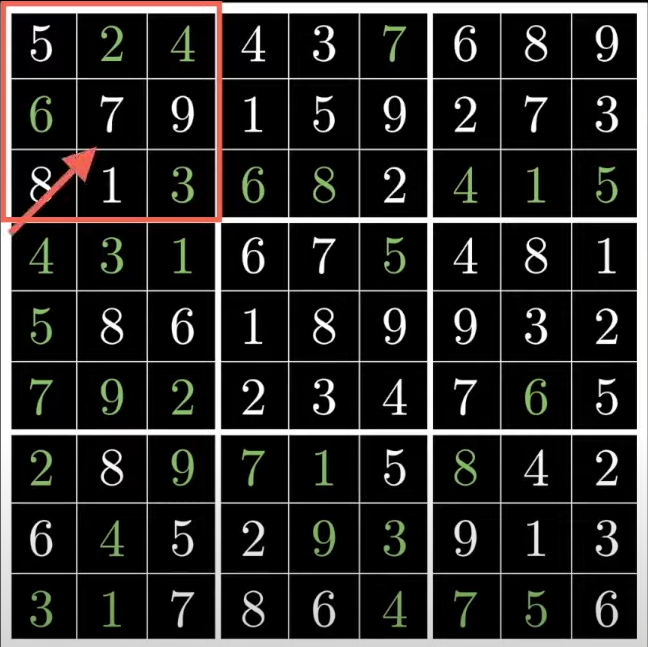
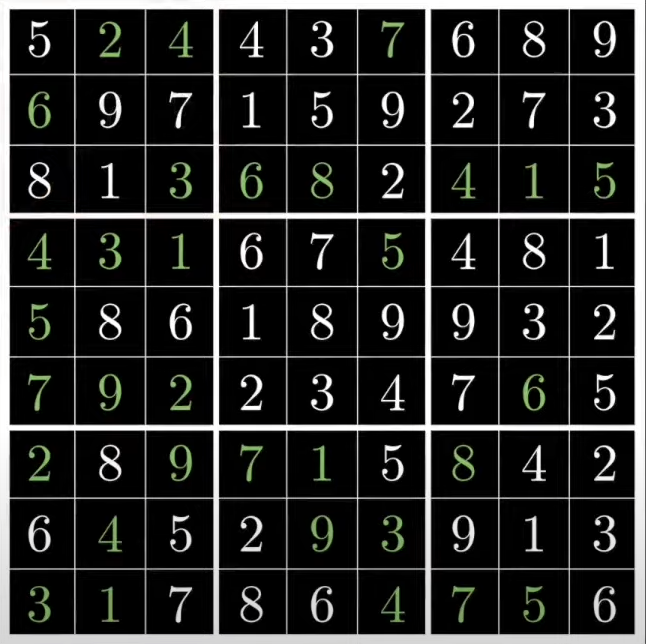
## Phân tích bài toán

* Sử dụng cấu trúc để biểu diễn trạng thái của bài toán. Một cấu trúc trạng thái Sudoku bao gồm
  + Mảng 2 chiều grid[9][9] chứa toàn bộ bảng Sudoku ban đầu (gồm các ô đã có sẵn dữ liệu và các ô trống có giá trị là 0).
  + Mảng 2 chiều fixedValueCells[9][9] chứa vị trí của những ô đã có sẵn dữ liệu cố định.
* Hướng giải quyết bài toán với giải thuật Mô phỏng luyện kim
  + Từ trạng thái Sudoku ban đầu, tiến hành điền ngẫu nhiên số từ 1-9 vào các ô trống sao cho mỗi khối 3x3 không có sự lặp lại.

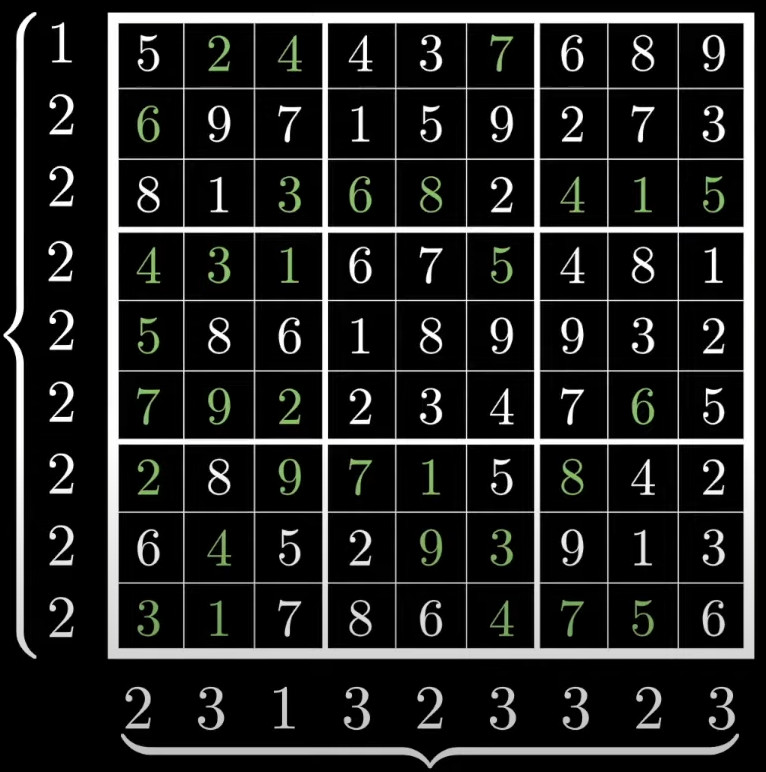
Hình 8 Sudoku trước và sau khi điền ngẫu nhiên số

* + Sinh ra trạng thái mới bằng cách chọn ngẫu nhiên 2 ô trong 1 khối bất kỳ và đổi giá trị của chúng với nhau

**** 

Hình Đổi vị trí 2 ô trong 1 khối bất kì để sinh trạng thái mới

* + Để biết trạng thái mới có tốt hơn không, chúng ta sẽ tính chi phí (cost) của các trạng thái bằng cách tính tổng số lượng của các giá trị bị trùng nhau trên các hàng và các cột và so sánh chúng với nhau



Hình Số lượng giá trị bị trùng trên hàng và cột (VD: cost = 39)

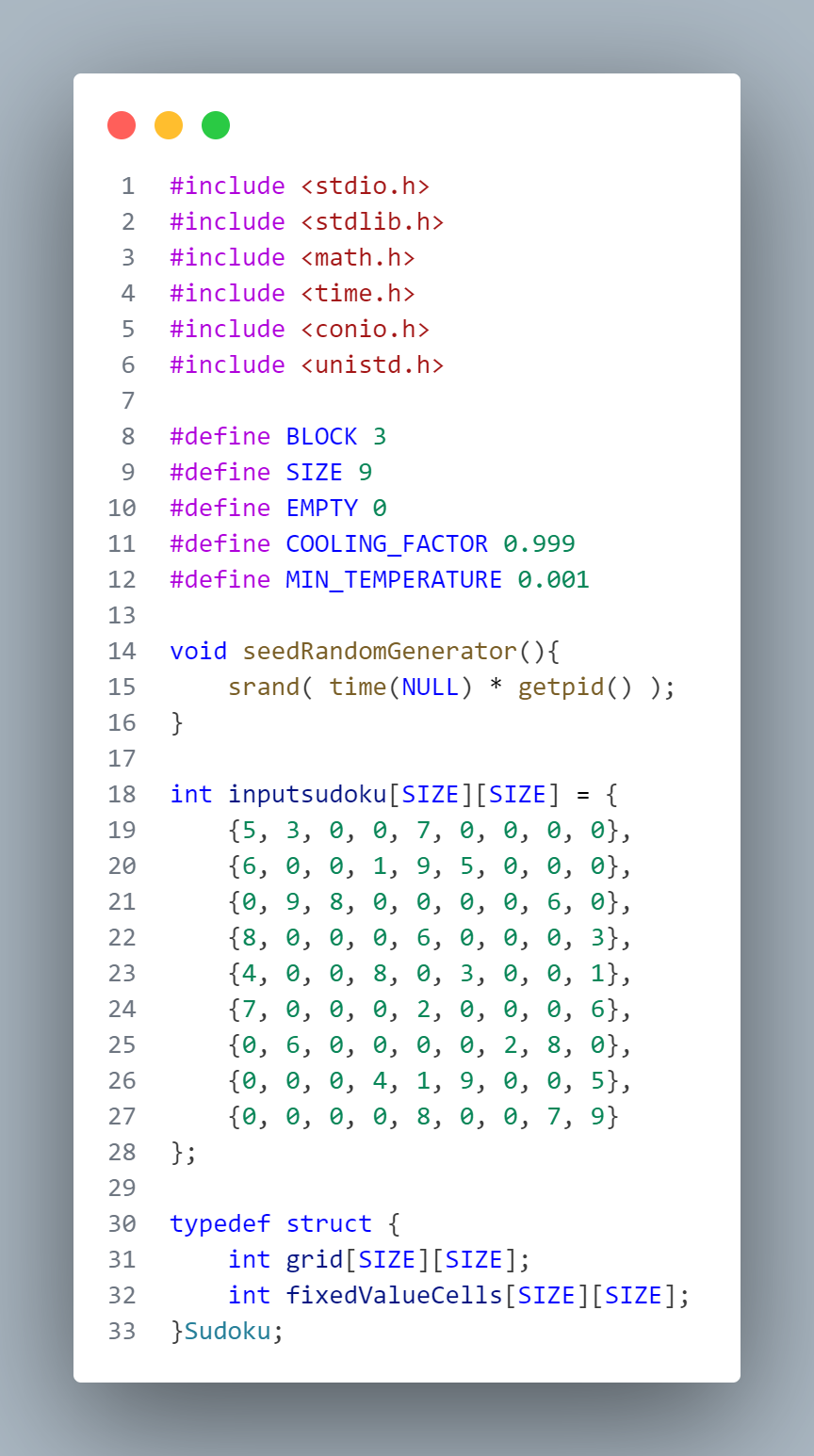
* + Chọn tham số kiểm soát T của giải thuật, chúng ta chọn bằng cách tính và lấy trung bình cost của 200 trạng thái ban đầu khi được điền ngẫu nhiên số.
  + Chọn tỷ lệ làm mát (COOLING\_RATE), ở đây chúng ta chọn COOLING\_RATE = 0.999.
  + Sau khi có những giá trị cần thiết, chúng ta tiến hành áp dụng giải thuật để giải bài toán.

Bảng so sánh kỹ thuật luyện kim và giải thuật SA áp dụng vào Sudoku:

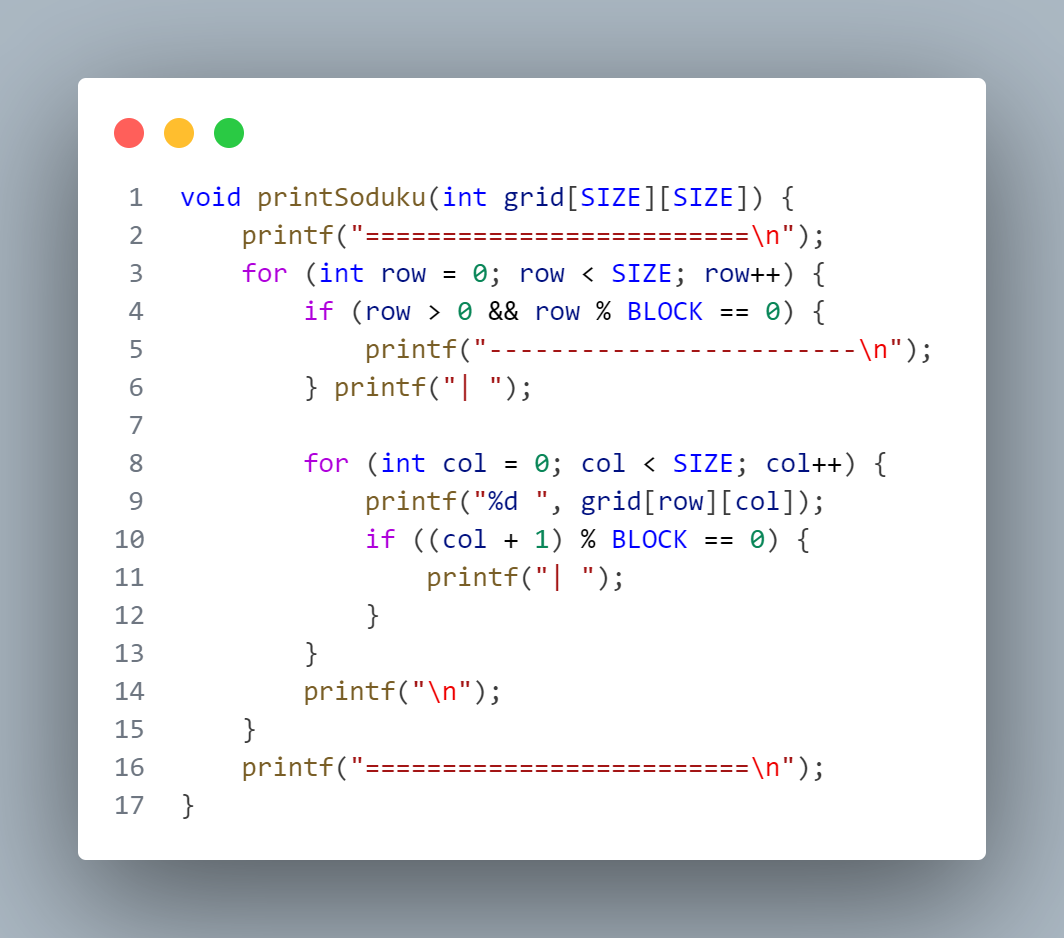
|  |  |
| --- | --- |
| **Physical** | **Sudoku Solving** |
| System state | Tập các trạng thái của khối Sudoku |
| Molecular Positions | Tập hợp biến  Các ô có dữ liệu không cố định |
| Energy | Hàm mục tiêu == Hàm tính chi phí |
| Minimizing Energy | Giảm chi phí về 0 |
| Ground State | Đáp án cuối cùng của Sudoku |
| Metastable state | Vùng tối ưu cục bộ:  Trạng thái của khối Sudoku đã được giải 1 phần (đúng khi so với hàng, cột, khối cục bộ) nhưng có thể dẫn đến đáp án sai cho cả bài toán |
| Quenching | Quá trình dẫn đến tìm ra tối ưu cục bộ: Với Sudoku, đây là quá trình điền vào những ô mà chỉ có 1 giá trị duy nhất thích hợp |
| Temperature | Tham số kiểm soát T Kiểm soát số lần lặp của giải thuật |
| Real Annealing | Mô phỏng luyện kim |

## Cài đặt giải thuật

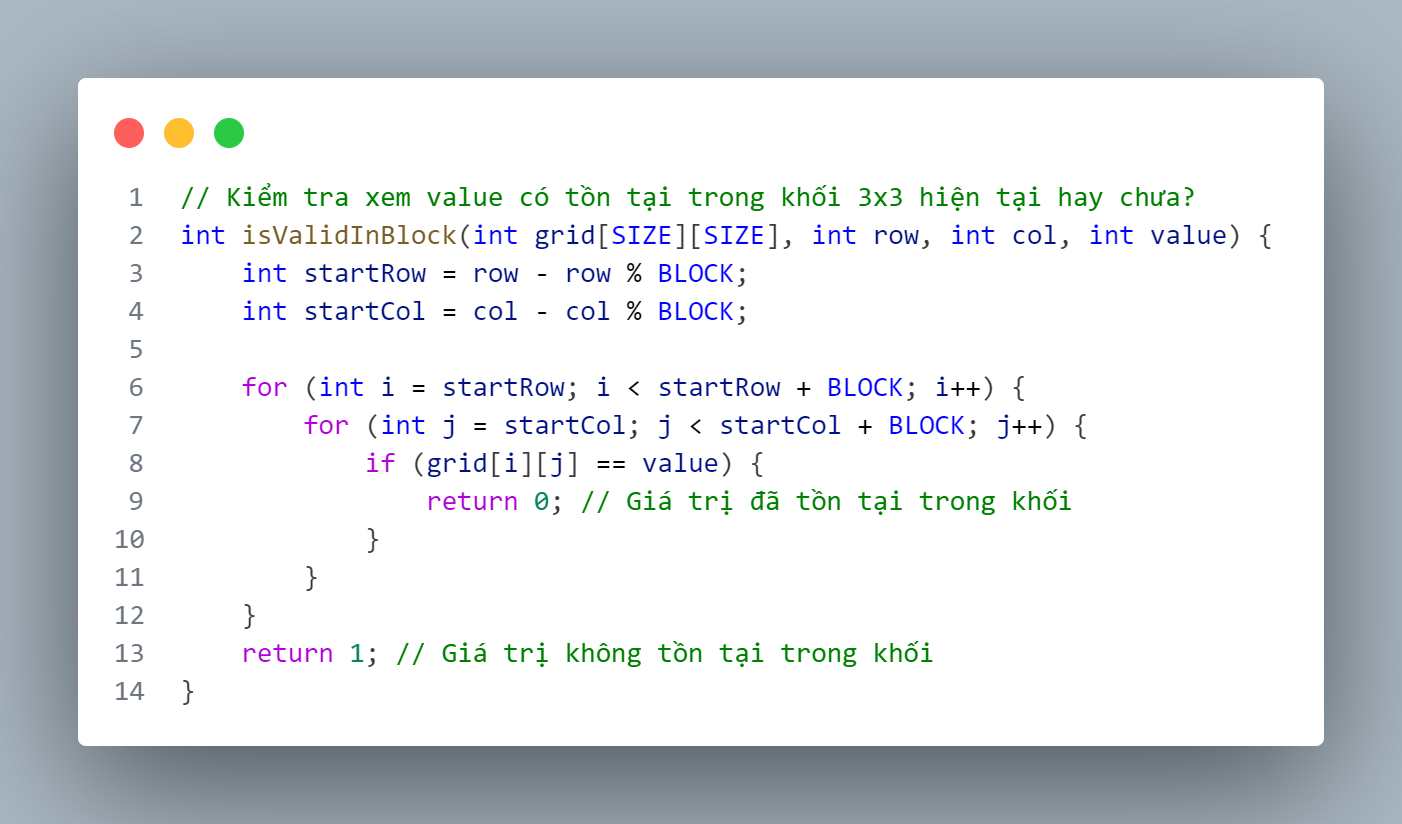
* Khai báo các biến cần thiết và cấu trúc Sudoku

****

* Hàm in Bảng Sudoku

****

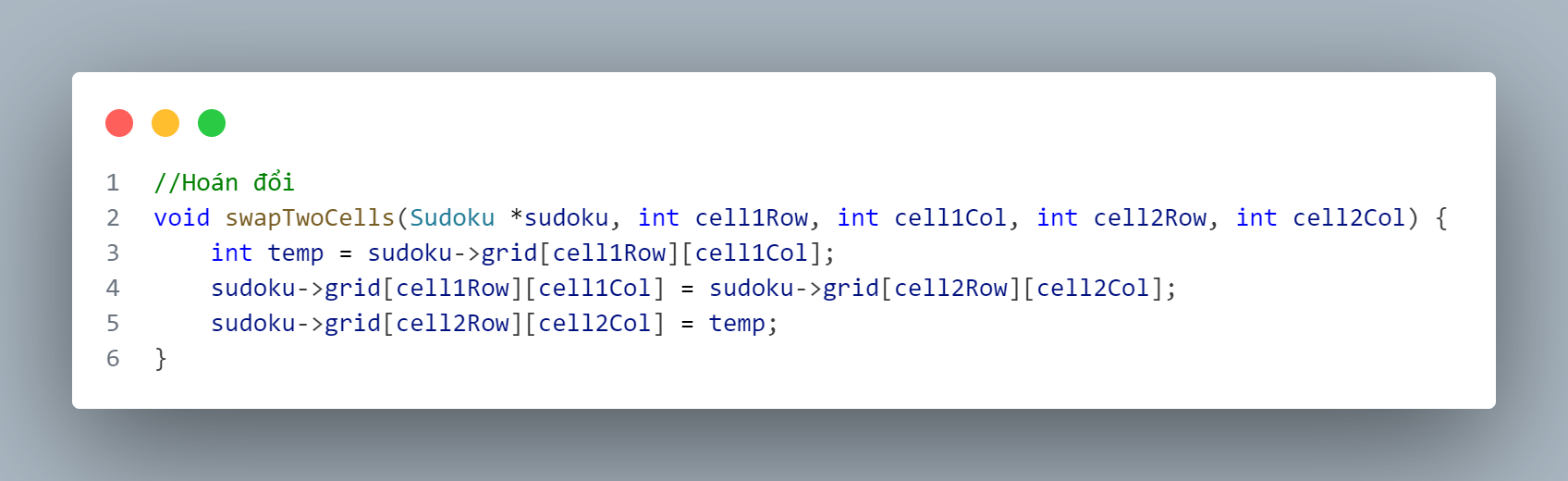
* Hàm kiểm tra xem giá trị có tồn tại trong khối con 3x3 đã cho hay chưa



* Hàm khởi tạo bảng Sudoku đầy đủ bằng cách điền các giá trị ngẫu nhiên từ 1-9 vào các khối con 3x3 sao cho không bị lặp lại



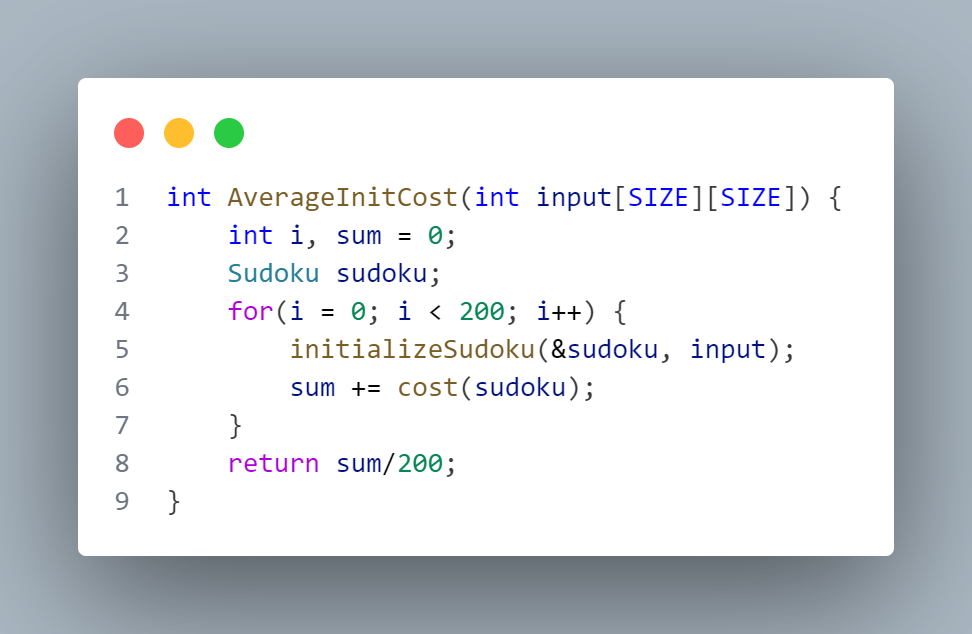
* Hàm hoán đổi giá trị của 2 ô



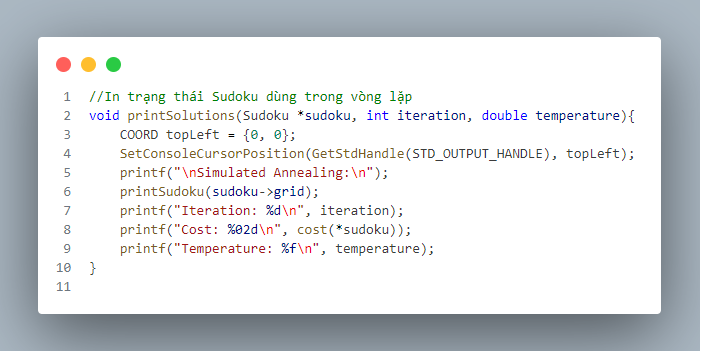
* Hàm tính chi phí (cost) trả về tổng số giá trị bị trùng lặp trên các hàng và cột



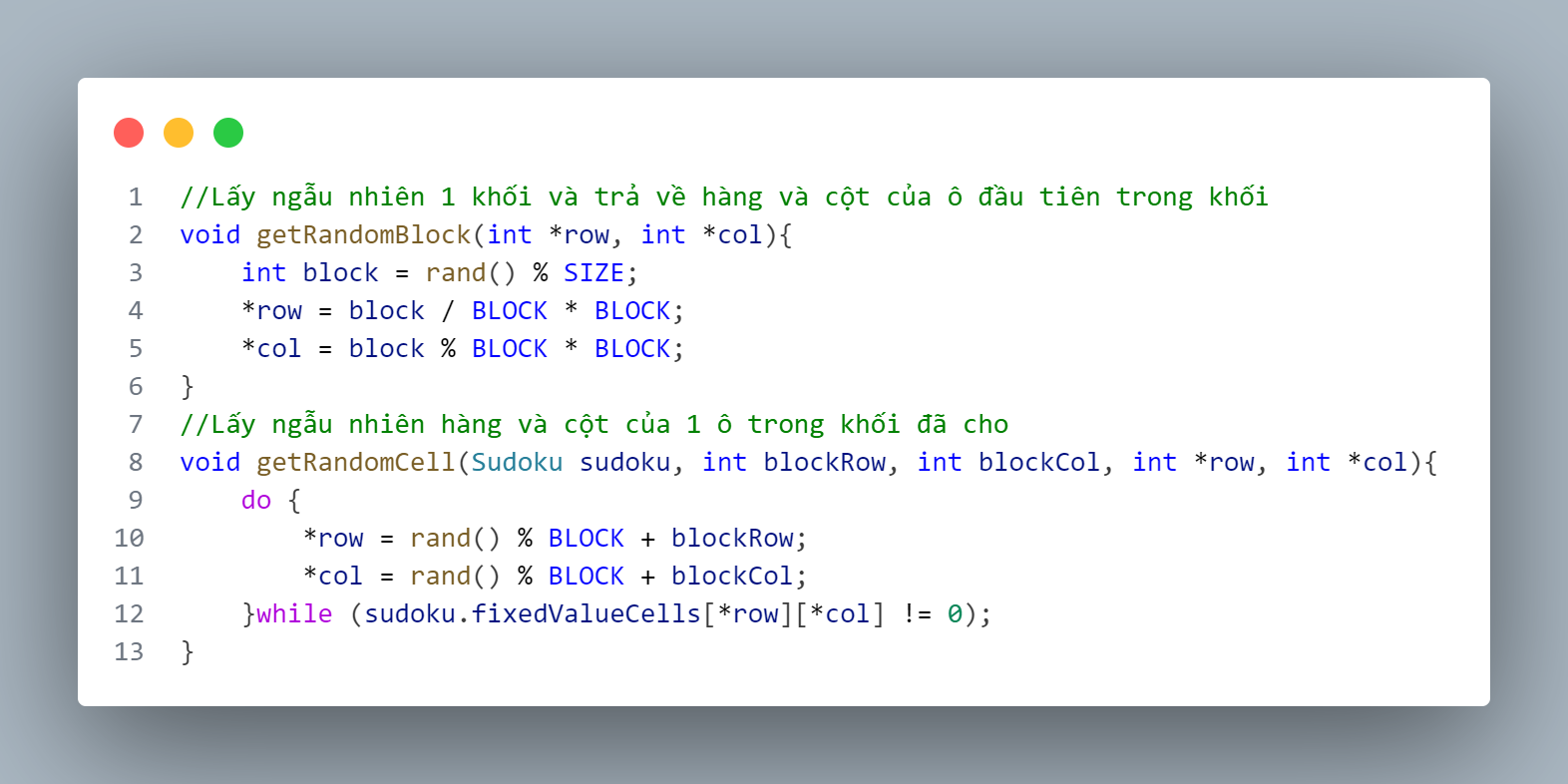
* Hàm tính tham số T bằng cách tính trung bình cost của 200 trạng thái sudoku ban đầu



* Hàm in trạng thái Sudoku và các đơn vị liên quan được dùng trong vòng lặp



* Hàm lấy ngẫu nhiên 1 khối 3x3 và 1 ô ngẫu nhiên trong khối đã cho



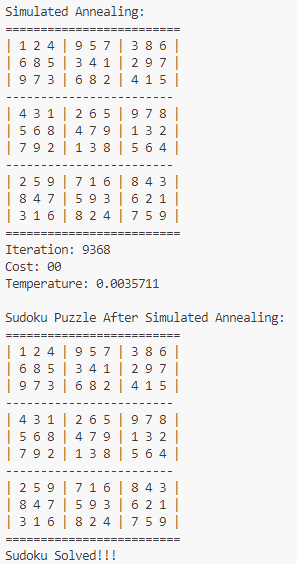
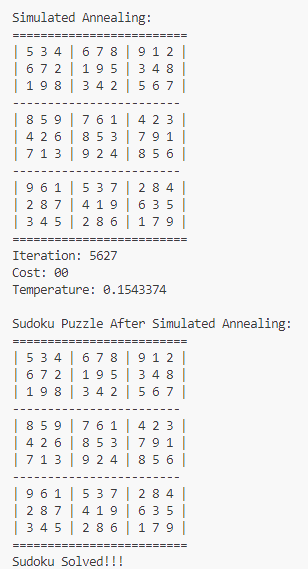
* Giải thuật Mô Phỏng luyện kim



* Hàm main



* Output: đáp án của Sudoku

**** ****

# PHẦN KẾT LUẬN

## Kết quả đạt được

* Đã áp dụng tương đối thành công giải thuật Mô phỏng luyện kim để giải bài toán Sudoku.

## Hướng phát triển

* Tinh chỉnh các giá trị làm mát (Cooling Rate) và nhiệt độ ban đầu (T) để quá trình tìm kiếm trở nên tối ưu hơn.
* Tối ưu cách viết code để độ phức tạp thấp hơn, gọn gàng và dễ hiểu các phép toán hơn.
* Có thể chuyển code sang ngôn ngữ khác như Python, C++.

## Hạn chế

* Do Simlated Annealing là giải thuật dựa trên xác suất nên sẽ **không** đảm bảo tìm ra được đáp án cuối cùng của bài toán Sudoku.

# TÀI LIỆU THAM KHẢO

1. Giáo trình Trí tuệ nhân tạo – CT332 (Trường CNTT&TT – Khoa KHMT)
2. Simulated Annealing Explained By Solving Sudoku - Artificial Intelligence (Youtube: Challenging Luck) <https://youtu.be/FyyVbuLZav8?si=C826LGvsIm7tdSV2>
3. Chat OpenAI: <https://chat.openai.com/>