**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**SVTH : Nguyễn Hoàng Gia**

**MSSV : 16110058**

**SVTH : Trần Huy Hào**

**MSSV : 16110063**

**GVHD: ThS.  Trần Công Tú**

**TP. Hồ Chí Minh, tháng 12 năm 2017**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**SVTH : Nguyễn Hoàng Gia**

**MSSV : 16110058**

**SVTH : Trần Huy Hào**

**MSSV : 16110063**

**GVHD: ThS.  Trần Công Tú**

**TP. Hồ Chí Minh, tháng 12 năm 2017**

Bảng 1:Bảng phân công nhiệm vụ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Nguyễn Hoàng Gia** | **Trần Huy Hào** | **Ghi chú:** |
| **1** | Thuật toán giải | 🗹 | 🗹 | 50/50 |
| **2** | Thiết kế Form | 🗹 | 🗹 | 50/50 |
| **3** | Tạo đề bài | 🗹 |  |  |
| **4** | Chức năng Trợ giúp | 🗹 |  |  |
| **5** | Remove | 🗹 |  |  |
| **6** | Viết báo cáo | 🗹 |  |  |
| **7** | Save player |  | 🗹 |  |
| **8** | Continue |  | 🗹 |  |
| **9** | Test |  | 🗹 |  |

# Mục Lục

[Danh mục các hình 1](#_Toc501046091)

[Lời Mở Đầu 2](#_Toc501046092)

[Chương 1. Giới thiệu 3](#_Toc501046093)

[*1.1.* *Giới thiệu đề tài* 3](#_Toc501046094)

[*1.2.* *Phân tích yêu cầu* 4](#_Toc501046095)

[Chương 2. Mô tả chức năng và xử lý. 5](#_Toc501046096)

[*2.1.* *Yêu cầu của người dùng:* 5](#_Toc501046097)

[*2.2.* *Chức năng của phần mềm*: 5](#_Toc501046098)

[*2.3.* *Mô hình xử lý:* 5](#_Toc501046099)

[Chương 3. Phân tích và thiết kế thuật toán 6](#_Toc501046100)

[*3.1.* *Ý tưởng thuật toán:* 6](#_Toc501046101)

[*3.2.* *Xây dựng thuật toán:* 7](#_Toc501046102)

[*3.3.* *Lưu đồ thuật toán:* 8](#_Toc501046103)

[Chương 4. Thiết kế giao diện 9](#_Toc501046104)

[*4.1.* *FormMain* 9](#_Toc501046105)

[*4.2.* *FormLevel* 10](#_Toc501046106)

[*4.3.* *FormPlay* 11](#_Toc501046107)

[*4.4.* *FormInfo* 12](#_Toc501046108)

[*4.5.* *FormScore* 12](#_Toc501046109)

[Chương 5. Cài đặt chương trình và kiểm thử 13](#_Toc501046110)

[*5.1.* *Cài đặt:* 13](#_Toc501046111)

[*5.2.* *Kiểm tra:* 13](#_Toc501046112)

[Chương 6. Đánh giá và hướng phát tiển 13](#_Toc501046113)

[*6.1.* *Những phần đã thực hiện* 13](#_Toc501046114)

[6.1.1 Ưu điểm: 13](#_Toc501046115)

[6.1.2 Nhược điểm: 13](#_Toc501046116)

[6.1.3 Ấn tượng: 13](#_Toc501046117)

[6.1.4 Khó khăn: 14](#_Toc501046118)

[*6.2.* *Hướng phát triển* 14](#_Toc501046119)

[Chương 7. Kết luận 14](#_Toc501046120)

[Link Github: 14](#_Toc501046121)

[Tài liệu tham khảo 15](#_Toc501046122)

# Danh mục các hình

[Hình 1.1.1: Demo game Sudoku 3](file:///D:\gia\CTDL\DA.docx#_Toc501046123)

[Hình 1.2.1: Cấp độ khó nhất của game 4](file:///D:\gia\CTDL\DA.docx#_Toc501046124)

[Hình 3.3.1: Lưu đồ thuật toán. 8](file:///D:\gia\CTDL\DA.docx#_Toc501046125)

[Hình 4.1.1: Giao diện chính chương trình 9](file:///D:\gia\CTDL\DA.docx#_Toc501046126)

[Hình 4.2.1: Chọn level chơi 10](file:///D:\gia\CTDL\DA.docx#_Toc501046127)

[Hình 4.3.1: Play Game 11](file:///D:\gia\CTDL\DA.docx#_Toc501046128)

[Hình 4.4.1: Nhập info player 12](file:///D:\gia\CTDL\DA.docx#_Toc501046129)

[Hình 4.5.1: Score 12](file:///D:\gia\CTDL\DA.docx#_Toc501046130)

# Lời Mở Đầu

Cấu trúc dữ liệu và giải thuật là học phần quan trọng đối với lập trình viên. Môn học này được xem là nền tảng của lập trình phần mềm. Nó là cơ sở vững chắc để giải quyết một số bài toán, đông thời cung cấp cho chúng ta hiểu biết về các thuật toán tác đông lên dữ liêu, cũng như tổ chức dữ liêu để giải quyết các bài toán sao cho hiểu quả tối ưu.

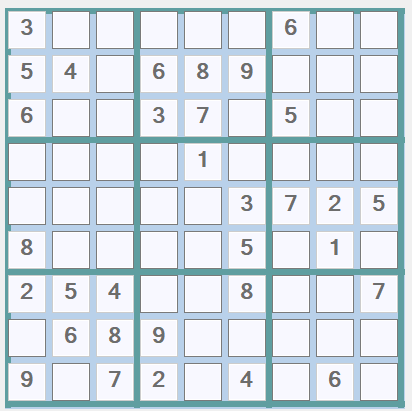
Sau học phần, nhóm chúng tôi đã nghiên cứu và thực hiện đồ án này như là một cách cũng cố và mở rộng kiến thức đồng thời ứng dụng vào thực tế. Thông qua quá trình làm đồ án, nhóm chúng tôi đã nắm bắt được kỹ thuật quan trọng trong việc xây dựng cấu trức dữ liệu và khử đệ quy bằng Stack nhằm tăng tốc độ chương trình.

Trò chơi Sudoku mà nhóm chúng tôi nghiên cứu và trình bày trong báo cáo sau đây là một ứng dụng điển hình của việc khử đệ quy.

# Giới thiệu

## *Giới thiệu đề tài*

Sudoku là một trò chơi trí tuệ nổi tiếng, thu hút nhiều người tham gia đặc biệt là giới trẻ. Ra đời ở Nhật và không lâu sau đã trở nên cực kỳ phổ biến trên thế giới. Quy luật của trò chơi tương đối đơn giản, cho một bàn hình vuông được chia thành một lưới 81 ô nhỏ trên 9 hàng và 9 cột. 81 ô nhỏ đó lại được chia thành 9 vùng, mỗi vùng có 9 ô. Đề bài Sudoku là một bàn hình vuông như thế, trên đó tại một số ô, người tạo đề đã điền sẳn một số giá trị.



Hình 1.1.1: Demo game Sudoku

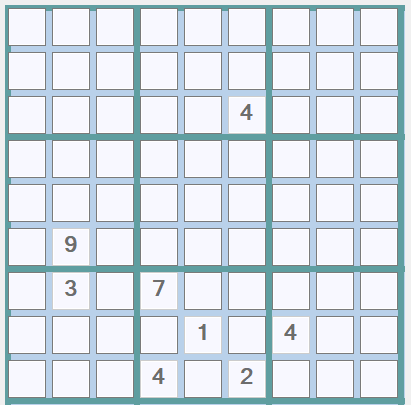
Nhiệm vụ của trò chơi là sử dụng các số từ 1 đên 9 để điền nốt vào các ô còn lại sao cho trên mỗi hàng, mỗi cột và mỗi vùng 9 ô, phải điền đầy đủ 9 số từ 1 đến 9.

Điều kiện của bài toán: 1 số được xem là điền đúng khi.

* Trong hàng ngang 9 ô của số đó, không có số nào lặp lại số này.
* Trong hàng dọc 9 ô của số đó, không có số nào lặp lại số này.
* Trong ô vuông 3x3 của số đó, không có số nào lặp lại số này.

## *Phân tích yêu cầu*

Ta nhận ra rằng với mỗi ô trống vị trí k trên lưới ta sẽ có nhiều phương án khác nhau và nó sẽ ảnh hưởng đến các vị trí k+1,k+2,… trên lưới ,có thể tìm tiếp phương án và dẫn đến kết quả cuối cùng và có những phương án sẽ không cho ra được kết quả và khi đó phải thực hiện khả năng khác của ô tại vị trí k và chương trình sẽ thực hiện rất lâu.



Hình 1.2.1: Cấp độ khó nhất của game

Đa số thuật toán giải Sudoku được thực hiện bằng đệ quy vì chương trình sẽ trong sáng dể hiểu (tùy từng trường hợp)…,.Tuy nhiên nếu gặp những bài toán khó , sẽ gọi hàm đệ quy rất nhiều lần, điều đó làm chương trình chậm và sẽ khó mở rộng lên các bài toán khó hơn 16x16,25x25….

Để khắc phục được điều đó nhóm em sẽ cài đặt bằng phương pháp khử đệ quy dựa trên giải thuật quay lui.

# Mô tả chức năng và xử lý.

## *Yêu cầu của người dùng:*

* Giao diện phải trực quan, đơn giản, không nhiều hình ảnh gây khó nhìn.
* Trò chơi phải dể sử dụng, dể nhớ. Khi người dùng bắt đầu trò chơi chỉ cần nhìn vào các biểu tượng có thể đoán được chức năng để chơi một cách dể dàng.
* Trò chơi phải đáp ứng được những mức độ của người chơi khác nhau như: dể, trung bình, khó.

## *Chức năng của phần mềm*:

* Chọn level.
* Chọn đề bài từ ngoài vào.
* Tự tạo đề bài có nghiệm.
* Lưu trạng thái.
* Lưu thông tin người chơi.
* Continue (tiếp tục trò chơi khi mở chương trình).
* Báo lỗi.
* Gợi ý.
* Hiển thị thời gian.
* Giải.
* Remove (xóa các số đã điền).
* Thoát.

## *Mô hình xử lý:*

* Chương trình hoạt động dựa trên đối tượng Sudoku và class Stack
  + Stack: Dùng để lưu vị trí cho việc quay lui theo quy tắc hoại động vào sau – ra trước (LIFO). Vì sử dụng ngôn ngữ C# nên nhóm em sẽ cài đặt Stack bằng mảng cho code đơn giản:
    - Push(data): Đưa dữ liệu vào Stack.
    - Pop(): Loại bỏ và trả giá trị phần tử Top của Stack.
    - isEmpty(): Kiểm tra Stack rỗng hay chưa.
  + Sudoku:
    - createSDK(): Tạo đề bài sudoku một cách ngẫu nhiên và đáp ứng yêu cầu phải tồn tại ít nhất 1 đáp án, để làm được việc này cần sử dụng 2 phương thức con:
      * khoitao(): Tạo ra đề bài chuẩn(giải đầy đủ các ô)
      * ran(): Hoán vị ngẫu nhiên 2 vị trí trong từng vùng 3x3
    - Remove():Xóa bỏ ngẫu nhiên số lượng ô phụ thuộc vào level người dùng chọn
    - timocuoi(): Khi có được đề bài hoàn chỉnh, tìm ô trống cuối cùng trong đề bài
    - timodau(): tìm vị trí ô đầu tiên để xét trường hợp ngoại lệ (Không có nghiệm).
    - DemSoLuong(): Đếm số lượng ô trống trong đề bài.
    - Solution():Giải đề bằng thuật toán quay lui
* Các class hổ trợ khác:
  + Business object: Làm việc với file
    - WriterPlayers(): Lưu thông tin người chơi xuống file
    - ReadPlayers(): Đọc dữ liệu người chơi từ file
    - Ghimang(): Ghi bàn chơi xuống file
    - DocMang(): Đọc mảng continues lên để tiếp tục chơi
    - Delete(): Xóa file continues cũ khi new game

# Phân tích và thiết kế thuật toán

## *Ý tưởng thuật toán:*

* Ý tưởng chính dựa vào phương pháp quay lui là các bước hướng tới lời giải cuối cùng dựa trên việc thử và sai
* Tại mổi bước :
  + Nếu có 1 lựa chọn được chấp nhận thì ghi nhận lại lựa chọn này và tiến hành các bước thử kế tiếp
  + Nếu tất cả lựa chọn không được chấp nhận thì trở lại bước trước, xóa bỏ ghi nhận khả năng và chọn lựa khả năng tiếp theo

## *Xây dựng thuật toán:*

Xem lưới đề bài là mảng 1 chiều thông qua giá trị k đại diện cho vị trí ô

Chương trình sẽ duyệt từng ô trên mảng.

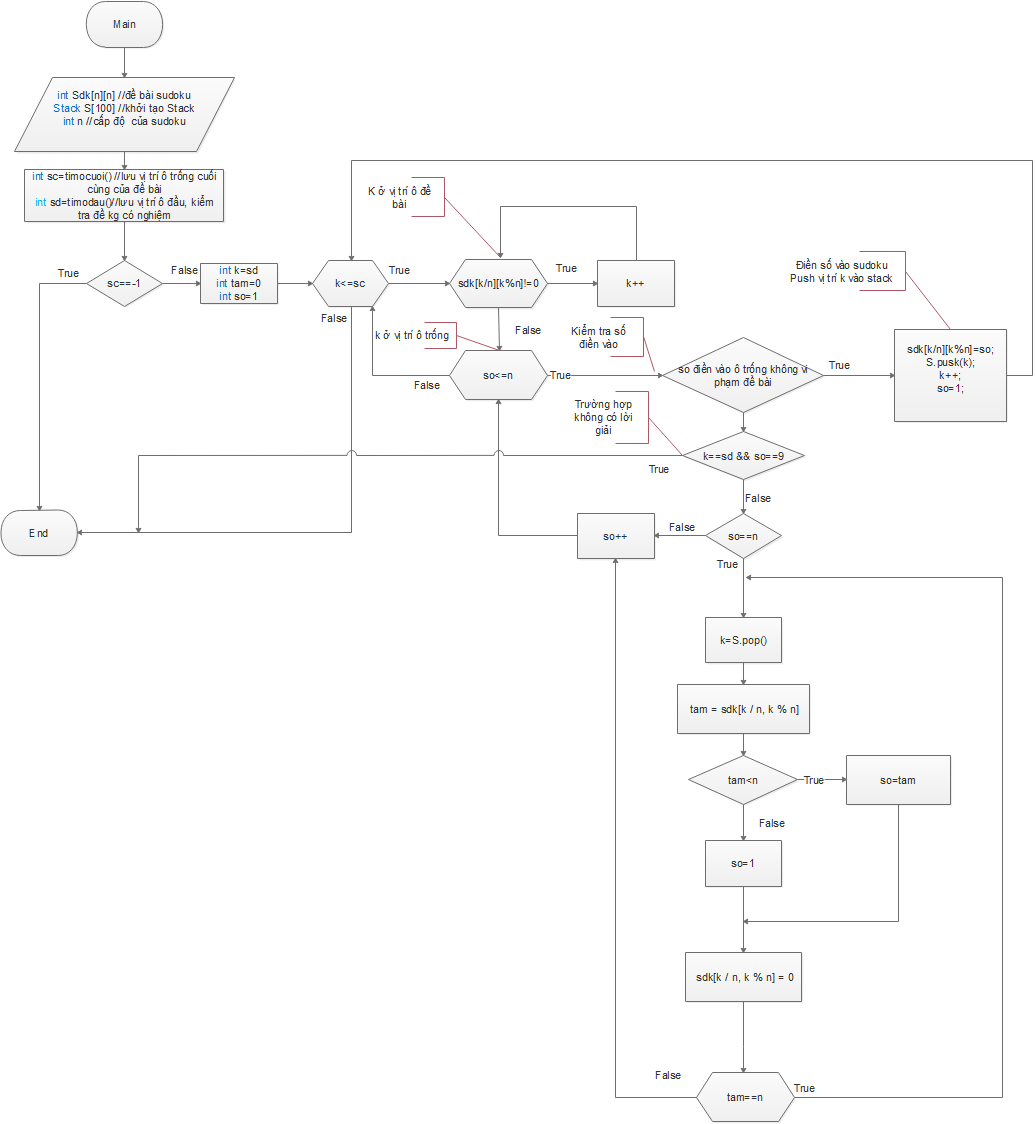
* Nếu ô đó là ô đề bài thì sẽ bỏ qua tiếp tục xét các ô tiếp theo (tăng k)
* Nếu là ô trống, tiến hành xét các giá trị số để điền (so) lần lược từ 1 đến 9 nếu có số thỏa điều kiện của bài toán thì gán giá trị số vào mảng (M[k]=so) tại vị trí k và đưa vị trí ô đó vào Stack (push(k)) và tiếp tục xét các ô tiếp theo.

Các trường hợp có thể xảy ra:

* Tại vị trí ô k, chúng ta xét hết tất cả các trường hợp có thể có mà vẫn không thỏa điều kiện bài toán:
  + Nếu Stack đang rỗng ta có thể kết luận lưới không có lời giải
  + Stack hiện tai không rỗng ta tiến hành lấy vị trí ô trống phía trước ra (quay lui) bằng cách pop() Stack và lưu vào k. Tiến hành kiểm tra giá trị tai ô thứ k:
    - Nếu nó bằng cấp độ của lưới (M[k]==9) (ví dụ bài toán ở đây 9x9) thì gán giá trị ô thứ k=0 (M[k]=0) giá trị so = 1 lặp lại viêc pop() Stack.
    - Khi nó không bằng (M[k]<9) gán giá trị (so=M[k]) và gán giá trị ô thứ k=0 (M[k]=0).

Điều kiện để chương trình hoàn thành là khi vị trí k trùng với ô trống cuối cùng trên mảng.

## *Lưu đồ thuật toán:*



Hình 3.3.1: Lưu đồ thuật toán.

# Thiết kế giao diện

## *FormMain*



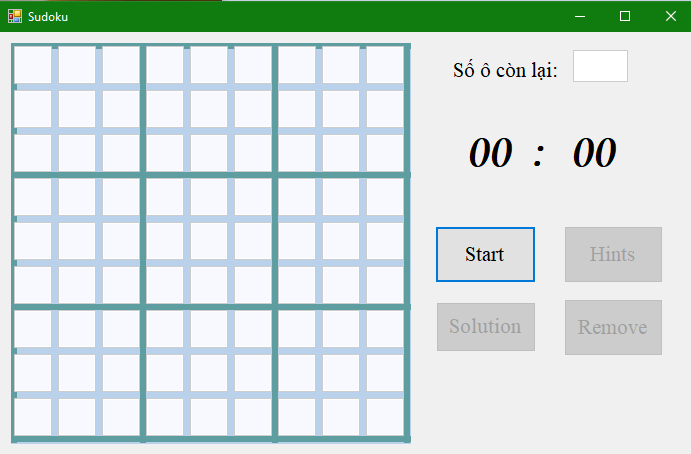
Hình 4.1.1: Giao diện chính chương trình

* New Game: Mở Form Level, cho người dùng chọn level game.
* Continue: Nếu có tồn tại file lưu, tiền hành đọc file lấy dữ liệu đề chơi cũ đổ dữ liệu vào FormPlay thông qua Constructor.
* Score: Hiện thông tin gồm tên và thời gian hoàn thành game của những người chơi gần nhất.

## *FormLevel*

Hình 4.2.1: Chọn level chơi

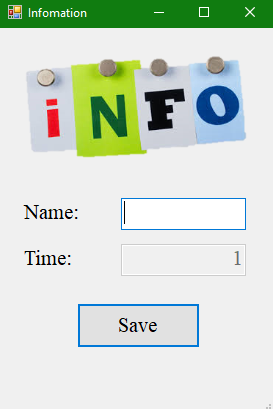
## *FormPlay*



Hình 4.3.1: Play Game

* Start: Bắt đầu trò chơi điền các ô đề bài vào Form đồn thời tắt chức năng Start, hiện các chức năng gợi ý, xóa hết, giải. Đồng thời tiếng hành sử dụng thuật toán đã cài đặt để giải sudoku và lưu vào mảng đáp án.
* Solution: Lắp đầy các ô sudoku.
* Hints: Gợi ý cho người chơi.
* Remove: Xóa hết các ô do người dùng điền.

## *FormInfo*



Hình 4.4.1: Nhập info player

* Khi đề sudoku được giải chính xác, tiến hành lưu dữ liệu người chơi.

## *FormScore*

Hình 4.5.1: Score

* Hiện thời gian chơi của người chơi cũ.

# Cài đặt chương trình và kiểm thử

## *Cài đặt:*

* Yêu cầu: .Net Framword 4.5.2
* Hệ điều hành: Windown

## *Kiểm tra:*

* Đề bài đúng, chương trình hoạt động các chức năng bình thường.
* Đề bài sai, chương trình thông báo lỗi và đóng chương trình.

# Đánh giá và hướng phát tiển

## *Những phần đã thực hiện*

### Ưu điểm:

* Tạo được đề bài ngẫu nhiên có nghiệm.
* Có thể lựa chọn đề bài tự tạo hoặc phần mềm tự sinh.
* Tốc độ giải nhanh trong bất kì trường hợp nào.
* Có thể giải từ đề bài trống hoàn toàn.

### Nhược điểm:

* Thuật toán giải vẫn chưa thật sự tối ưu, ở mỗi bước đều xét các khả năng từ 1 đến 9.
* Chưa áp dụng cho các cấp độ cao hơn hoặc thấp hơn.
* Thuật toán giải khá dài và rắt rối hơn so với sử dụng phương pháp đệ quy

### Ấn tượng:

* Khả năng giải nhanh và trên mặt ý tưởng có thể áp dụng thuật toán cho các bài toán khác như: Quân hậu, mã đi tuần với mức độ khó hơn.
* Cách sinh ra đề bài ngẫu nhiên.

### Khó khăn:

* + Trong quá trình làm nhóm chúng tôi vẫn gặp một số khó khăn về mặt thiết kế Form xử lý 1 số các event phức tạp như chỉ cho người chơi nhập số, không nhận bất kì các ký tự khác.

## *Hướng phát triển*

* Mở rộng cho các cấp độ 4x4, 16x16, 25x25.
* Cài đặt lại thuật toán giải bằng việc phân tích hàng dọc, hàng ngang, và vùng 3x3 tại ô cần xét để loại rút ngắn khả năng có thể có, tiết kiệm thời gian hơn.
* Phát triển ứng dụng trên các hệ điều hành khác.

# Kết luận

Thông qua việc tìm hiểu về ứng dụng của Stack để viết chương trình giải Sudoku nhóm chúng tôi đã rút ra được nhiều điều. Cùng một mục đích giải nhưng lại có nhiều thuật toán khác nhau, khác phục được những hạn chế của đệ quy.

Từ cách xác định thuật toán đến tư duy logic và chương trình đã chạy thành công trên phần mềm C#. Đồng thời biết được cách làm bài một bài báo cáo đúng chuẩn.

# Link Github: [Link](https://github.com/gia01020304/Solution-Sudoku.git)

# Tài liệu tham khảo

1. Sudoku Generation and Difficulty Metrics, *Thuật toán tạo random thỏa điều kiện,* [*http://www.mathspace.com/COMAP/Training\_Materials/Team2975\_ProblemB.pdf*](http://www.mathspace.com/COMAP/Training_Materials/Team2975_ProblemB.pdf)*..*
2. Randomized Algorithms, Dẫn nhập thuật toán ngẫu nhiên, <http://dkhoa.me/post/blog/intro_randalgs/>.