

**ĐẠI HỌC ĐÀ NẴNG
ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT
ĐỀ TÀI :VIẾT CHƯƠNG TRÌNH TRÒ CHƠI SUDOKU

GIẢNG VIÊN HƯỚNG DẪN: Phan Chí Tùng

SINH VIÊN THỰC HIỆN:

Ngô Tấn Trí

LỚP: 17T3 NHÓM: 17.77A

Trịnh Quang Duẩn

LỚP: 17T3 NHÓM: 17.77A

Đà Nẵng : 02-01-2019

LỜI MỞ ĐẦU

Cấu trúc dữ liệu và phân tích thiết kế giải thuật là hai học phần rất quan trọng đối với người lập trình . Học phần này được xem là nền tảng của lập trình máy tính. Nó là cơ sở vững chắc để giải quyết nhiều bài toán, đồng thời cung cấp cho ta sự hiểu biết về các giải thuật tác động lên dữ liệu , cũng như cách tổ chức dữ liệu để tối ưu hóa bài toán.

Sau khi học xong học phần lý thuyết, nhóm chúng em đã nghiên cứu và thực hiện đồ án này cũng như là một cách để củng cố và mở rộng kiến thức. Thông qua quá trình thực hiện đồ án , chúng em đã nắm bắt được kỹ thuật quan trọng của việc xây dựng cấu trúc dữ liệu và phân tích thiết kế giải thuật sao cho tối ưu nhất.

Bài toán “Viết chương trình trò chơi sudoku” mà chúng em nghiên cứu và trình bày trong báo cáo sau đây là một ví dụ.

Em xin chân thành cảm ơn thầy Phan Chí Tùng đã bổ sung ý tưởng và tận tình giúp đỡ chúng em thực hiện đồ án này. Em cũng rất mong nhận được sự góp ý từ phía thầy, cô để bài làm của chúng em được hoàn thiện hơn.

MỤC LỤC

LỜI MỞ ĐẦU.....	2
MỤC LỤC.....	3
DANH MỤC HÌNH VẼ.....	4
1. GIỚI THIỆU ĐỀ TÀI.....	5
2. CƠ SỞ LÝ THUYẾT.....	6
2.1. Ý tưởng.....	6
2.2. Cơ sở lý thuyết.....	6
3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN.....	7
3.1. Phát biểu bài toán.....	7
3.2. Cấu trúc dữ liệu.....	8
3.3. Thuật toán.....	8
4. CHƯƠNG TRÌNH VÀ KẾT QUẢ.....	9
4.1. Tổ chức chương trình.....	9
4.2. Ngôn ngữ cài đặt.....	11
4.3. Kết quả.....	11
4.3.1. Kết quả thực thi của chương trình.....	12
4.3.2. Nhận xét.....	12
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	13
5.1. Kết luận.....	13
5.2. Hướng phát triển.....	13
TÀI LIỆU THAM KHẢO.....	13

DANH MỤC HÌNH VẼ

Hình 1:Hình ảnh lưới ô vuông Sudoku kích thước 9x9.

Hình 2: Mô tả đề bài sudoku.

1. GIỚI THIỆU ĐỀ TÀI

1.1. Lịch sử trò chơi Sudoku

Sudoku có lịch sử từ hàng ngàn năm. Năm 990, một danh sách những “Ô số kỳ ảo” đã xuất hiện và tỏ ra không khác mấy so với bản Sudoku xuất hiện trong từ điển bách khoa Ikhwan Al-Salfa của các học giả người Ả Rập.

Abraham Ben Ibn-Ezra – một nhà triết học kiêm chiêm tinh học người Hispanic(Tây Ban Nha – Bồ Đào Nha) gốc Do Thái – bắt đầu quảng bá khối vuông buduh ở châu Âu. Ông đi khắp Tây Ban Nha, Ý và các nước khác ở châu Âu để giới thiệu với công chúng về “những ô số kỳ ảo”.

Ý tưởng tạo nên những ranh giới cho các khối vuông (biến nó thành trò chơi) đã được Ahmed Al-Buni ghi lại vào năm 1225, mặc dù phương pháp này được tin là có xuất xứ từ Ba-Tur.[1]

1.2 Sự phát triển của trò chơi Sudoku

Sudoku có thêm một bước tiến hóa mới vào năm 1776 khi một nhà toán học kiêm vật lý người Thụy Sĩ tên Leonhard Euler bắt đầu nghiên cứu và phát triển các luật chơi mà ngày nay ta gọi là luật chơi Sudoku.

Sudoku lần đầu tiên được xuất bản vào cuối thập niên 1970 trong một tờ tạp chí ở NewYork. Tờ tạp chí này đã giới thiệu về các ô số kỳ ảo và khuôn nó lại trong một lưới 9x9, tạo thành từ các khối 3x3. Và như thế, Sudoku đã ra đời.

Năm 1986, trong một chuyến đi Mỹ, một nhà xuất bản Nhật Bản, Nikoli, đã khám phá ra các ô số. Họ đặt tên cho nó là Sudoku (Su là số, Doku là đơn độc), và làm cho nó nhanh chóng trở thành một trò chơi phổ biến ở Nhật Bản.

Năm 2004, niềm đam mê Sudoku đã đưa Wayne Gould đến với London (Anh). Nhân một chuyến thăm ngẫu nhiên báo The Times, Gould đã thuyết phục tổng biên tập của báo này cho đăng Sudoku bên cạnh các ô chữ. Độc giả lập tức bị cuốn hút. Chỉ trong vài tuần lễ, các tờ báo trên khắp nước Anh đã thi nhau đăng Sudoku. Từ đó, Sudoku bắt đầu lan rộng sang Mỹ, Canada, Úc, Pháp, Nam Phi và nhiều quốc gia khác. [1]

2. CƠ SỞ LÝ THUYẾT

2.1. Hướng dẫn luật chơi

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Hình 1. Hình ảnh lưới ô vuông Sudoku kích thước 9x9.

Luật chơi: Điền vào những ô trống những con số thích hợp theo quy luật đơn giản:

- Các ô ở mỗi hàng (ngang) phải có đủ các số từ 1 đến 9, không cần theo thứ tự.
- Các ô ở mỗi cột (dọc) phải có đủ các số từ 1 đến 9, không cần theo thứ tự.
- Mỗi miền 3x3, được viền đậm, phải có đủ các số từ 1 đến 9
- Trò chơi bắt đầu với một lưới Sudoku, trong đó một số ô đã cho sẵn các con số đúng, phải suy luận để tìm ra những con số trong các ô trống còn lại.
- Các đề sudoku mức dễ thường bắt đầu với nhiều con số đã được điền sẵn, do đó sẽ dễ tìm ra đáp án hơn. Càng tìm ra nhiều con số, sẽ càng dễ tìm ra các con số khác

2.2. Cơ sở lý thuyết

Trong bài này, chúng tôi sẽ giới thiệu cách giải sudoku bằng thuật toán quay lui. Quay lui là một kỹ thuật thiết kế giải thuật dựa trên đệ quy. Ý tưởng của quay lui là tìm lời giải từng bước, mỗi bước chọn một số trong số các lựa chọn khả dĩ và đệ quy cho đến khi tìm được nghiệm của bài toán thì ta dừng lại. Ở đây, quay lui còn có thể giúp chúng ta liệt kê tất cả các nghiệm của bài toán. Mặc dù có những ưu điểm như vậy nhưng thuật toán quay lui có độ phức tạp lớn, chỉ giải quyết được với các bài toán có kích thước nhỏ.[2]

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1. Phát biểu bài toán

Đề bài: Có một hình vuông được chia thành 9x9 ô vuông con. Mỗi ô vuông con có giá trị trong khoảng từ 1 đến 9. Ban đầu hình vuông có một số ô vuông con cho trước (có điền sẵn số) và còn lại là trống (ta mặc định là số 0). Hãy điền các số từ 1 đến 9 vào các ô còn lại sao cho: Hàng ngang là các số khác nhau từ 1 đến 9, hàng dọc là các số khác nhau từ 1 đến 9. Và mỗi khối 3x3 chính là các số khác nhau từ 1 đến 9.

Ví dụ một câu đố và lời giải tương tự như sau:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Hình 2. Mô tả đề bài sudoku

Đầu vào (Input):

```

5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9
    
```

Đầu ra (Output):

```

5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9
    
```

3.2. Cấu trúc dữ liệu

Để thực hiện được chương trình giải mã bài toán Sudoku này, chúng tôi sử dụng những cấu trúc dữ liệu cơ bản đó là mảng hai chiều để lưu trữ và xử lý đầu vào của bài toán, ngoài ra chúng tôi còn sử dụng tệp để đọc và viết đáp án của bài toán.

3.3. Thuật toán

Ý tưởng của thuật toán như sau:

Mỗi bước tìm tập các giá trị khả dĩ để điền vào ô trống và sau đó đệ quy để điền ô tiếp theo, ở đây chúng tôi sử dụng thủ tục Feasible(s,x,y,k) để kiểm tra xem giá trị k có khả dĩ với ô s[x][y] không, và thuật toán được mô tả như sau:[2]

Phần 1: Hàm SUDOKU dùng để duyệt hết tất cả các ô (x,y) trên lưới 9x9 và điền những giá trị vào những ô vuông đó, đệ quy cho đến khi tìm được 1 nghiệm phù hợp và kết thúc.

```

SUDOKU(S[1, 2, ..., 9][1, 2, ..., 9], x, y):
    if y = 10
        if x = 9
            print S
        else
            SUDOKU(S[1, 2, ..., 9][1, 2, ..., 9], x + 1, 1)
    else if S[x, y] = ∅
        for k ← 1 to 9
            if FEASIBLE(S, x, y, k)
                S[x, y] ← k
                SUDOKU(S[1, 2, ..., 9][1, 2, ..., 9], x, y + 1)
                S[x, y] ← ∅    << for next branching >>
    else
        << S[x, y] is given >>
        SUDOKU(S[1, 2, ..., 9][1, 2, ..., 9], x, y + 1)
    
```

Phần 2: Hàm FEASIBLE mô tả quá trình kiểm tra 1 giá trị k từ 1 đến 9 có khả dĩ trong ô (x,y) của lưới hay không ?


```
FEASIBLE( $S[1, 2, \dots, 9][1, 2, \dots, 9], x, y, k$ ):  
  for  $i \leftarrow 1$  to 9  
    if  $S[x, i] = k$   
      return FALSE  
  for  $i \leftarrow 1$  to 9  
    if  $S[i, y] = k$   
      return FALSE  
   $a \leftarrow \lfloor (x - 1)/3 \rfloor, b \leftarrow \lfloor (y - 1)/3 \rfloor$   
  for  $i \leftarrow 3a + 1$  to  $3a + 3$   
    for  $j \leftarrow 3b + 1$  to  $3b + 3$   
      if  $S[i, j] = k$   
        return FALSE  
  return TRUE
```

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1. Tổ chức chương trình

Dưới đây là mã chương trình của bài toán :

```
#include<iostream>  
using namespace std;  
int dau;  
// In kết quả ra tệp  
void printSolution(int a[9][9]){  
    int i,j;  
    for(i=0;i<9;i++){  
        for(j=0;j<9;j++) cout<<a[i][j]<<" ";  
        cout<<"\n";  
    }  
}  
// Hàm kiểm tra giá trị k có khả dĩ trong ô (x,y) hay không  
int feasible (int s[][9],int x,int y,int k){  
    int i=0,j=0;  
    for(i=0;i<9;i++){  
        if(s[x][i]==k) return 0;  
    }  
    for(i=0;i<9;i++){  
        if(s[i][y]==k) return 0;  
    }  
}
```

```

int a=x/3,b=y/3;
for(i=3*a;i<3*a+3;i++){
    for(j=3*b;j<3*b+3;j++){
        if(s[i][j]==k) return 0;
    }
}
return 1;
}

// Tìm nghiệm của bài toán
void solve_sudoku(int s[][9],int x,int y){
    if(y==9){
        if(x==8){
            dau=1;
            printSolution(s);
            return ;
        }else{
            solve_sudoku(s,x+1,0);
        }
    }else if(s[x][y]==0){
        int k=0;
        for(k=1;k<=9;k++){
            if(feasible(s,x,y,k)){
                s[x][y]=k;
                solve_sudoku(s,x,y+1);
                s[x][y]=0;
            }
        }
    } else {solve_sudoku(s,x,y+1);
    }
}

int main(){

```

```
int i,j;
freopen("input.txt","r",stdin); // đọc tệp
freopen("output.txt","w",stdout); // in kết quả ra tệp
int a[9][9];
for(i=0;i<9;i++)
    for(j=0;j<9;j++) cin>>a[i][j];
solve_sudoku(a,0,0);
if(dau==0) cout<<"No solution!"; // Nếu vô nghiệm thì in ra No Solution
return 0;
}
```

4.2. Ngôn ngữ cài đặt

C/C++

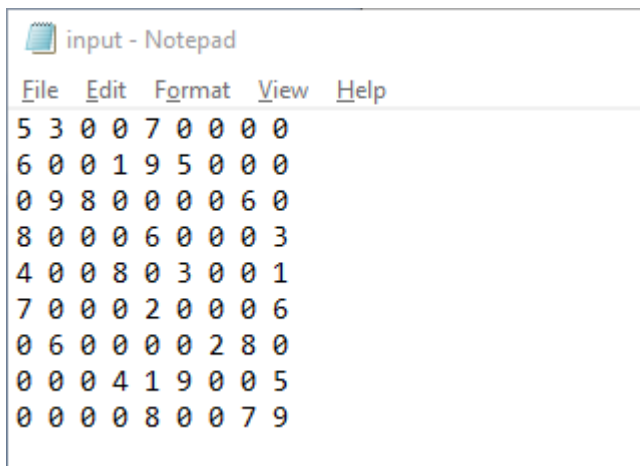
4.3. Kết quả

4.3.1. Kết quả thực thi của chương trình

Ứng với các đầu vào khác nhau thì kết quả sẽ cho những bài toán khác nhau.

Ví dụ1:

Input:



```
input - Notepad
File Edit Format View Help
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9
```

➔ Output:

```
output - Notepad
File Edit Format View Help
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9
```

Ví dụ 2:

Input:

```
input - Notepad
File Edit Format View Help
5 3 1 2 7 8 8 9 9
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9
```

➔ Output:

```
output - Notepad
File Edit Format View Help
No solution!
```

4.3.2. Nhận xét:

- + Không phải input nào cũng cho kết quả, tùy thuộc vào dữ kiện đầu vào mà bài toán đã cho có nghiệm hoặc vô nghiệm.
- + Nếu dữ kiện trong input đầu vào cho càng nhiều thì chương trình chạy nhanh hơn và ngược lại.
- + Vì ở đây chúng ta sử dụng thuật toán quay lui nên độ phức tạp khá lớn, chỉ có thể giải quyết tốt đối với các lưới ô vuông có kích thích nhỏ

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Chúng ta có thể rút ra những ưu và nhược điểm của thuật toán giải quyết bài toán này như sau:

- Ưu điểm :
 - + Thuật toán quay lui này đã giải quyết triệt để bài toán sudoku với lưới ô vuông kích thước 9x9
 - + Chương trình dễ cài đặt
- Nhược điểm:
 - + Thuật toán chỉ giải quyết được những bài toán có kích thước đầu vào nhỏ do độ phức tạp ở đây tới dạng lũy thừa.

5.2. Hướng phát triển

Trong tương lai, cùng với sự phát triển khoa học, chúng tôi sẽ cố gắng nghiên cứu để giảm độ phức tạp của thuật toán hoặc đưa ra những thuật toán tối ưu hơn để có thể giải quyết được những bài toán có kích thước lớn hơn

TÀI LIỆU THAM KHẢO

- [1]. <https://vi.wikipedia.org/wiki/Sudoku>
- [2]. <http://www.giaithuatlaptrinh.com/?tag=sudoku>