Со школьного возраста мне нравились судоку. Помогало коротать время в пути в школу (да и сейчас играю по дороги на работу). В скором времени я и бабушку смог подсадить на судоку, но проблема была в том, что она не могла играть на электронном устройстве. Потому пришла идея в голову сделать свой судоку, который можно будет распечатывать.

О том, как сделать свой генератор судоку и сколько оценочно различных вариантов может быть и пойдет речь под катом.

\* КАРТИНКА СУДОКУ \*

Реализовывать буду все на Pascal, так как он был ближе всех к мышке на момент написания статьи. Однако все можно реализовать и на других языках.

Способ первый. Алфавитная замена.

Первый способ, который мы рассмотрим, будет замена одних цифр на другие в исходном примера судоку. За исходный вариант возьмем следующую таблицу

\* КАРТИНКА БАЗОВОЙ ТАБЛИЦЫ \*

Суть способа состоит в том, чтобы каждой цифре сопоставить другую цифры и тем самым изменить всю судоку целиком.

Для этого воспользуемся одномерной матрицей с индексами от 1 до 9 и заполним в самом начале все ячейки в соответствии с их индексами. А дальше будем брать две случайные ячейки и менять их значения между собой. Заметим, что проделав данную операцию нечетное количество раз у нас не получится исходный вариант заполнения матрицы (доказательство предоставляется читателю). Для сомневающихся в конце программы проводится генерация одного миллиона матриц судоку и оценка их совпадений с исходным.

\* КАРТИНКА СООТВЕТСТВИЯ ЦИФРОВОГО АЛФАВИТА \*

Код программы на Pascal

**type** rec=**record**//алфавит замены

num:**array** [1..9] **of** integer;

**constructor** Create();

**var** i:integer;

**begin**

**for** i:=1 **to** 9 **do**

num[i]:=i;

**end**;

**procedure** rand();//случайный обмен двух чисел в алфавите

**var** i,j,k:integer;

**begin**

i:=random(1,9);

j:=i;

**while**(i=j) **do** j:=random(1,9);

k:=num[i];

num[i]:=num[j];

num[j]:=k;

**end**;

**procedure** wr();//вывод алфавита на экран

**var** i:integer;

**begin**

**for** i:=1 **to** 9 **do**

write(num[i]+' ');

writeln;

**end**;

**end**;

Подсчитав, получим 9! вариантов, а это 362 880 различных комбинаций.

Способ второй. Матричные перестановки.

Это способ, который был ещё на Хабре пару лет назад. Почитать подробности можно тут ССЫЛКУ ВСТАВЬ!!!!

А тут приведу краткую выжимку:

Можно поменять строки/столбцы в тройках, как показано на рисунке

\* РИСУНКИ СВАЙПОВ \*

\* ВТОРОЙ РИСУНОК СВАЙПОВ \*

Стоит так же отметить, что нечетное количество перестановок гарантирует отличие от первоначального варианта.

Потому реализуем сначала обмен строк и обмен столбцов в матрице судоку. Так же для упрощения кода вставим туда функцию замены цифр по алфавиту (способ, описанный ранее)

Код программы на Pascal

**type** sudoku=**record**//таблица судоку

num:**array** [1..9] **of array** [1..9] **of** integer;

**constructor** Create();//ряд от 1 до 9 сдвинутый циклично так, чтобы таблица соответствовала правилам судоку

**begin**

num[1][1]:=1; num[1][2]:=2; num[1][3]:=3; num[1][4]:=4; num[1][5]:=5; num[1][6]:=6; num[1][7]:=7; num[1][8]:=8; num[1][9]:=9;

num[2][1]:=4; num[2][2]:=5; num[2][3]:=6; num[2][4]:=7; num[2][5]:=8; num[2][6]:=9; num[2][7]:=1; num[2][8]:=2; num[2][9]:=3;

num[3][1]:=7; num[3][2]:=8; num[3][3]:=9; num[3][4]:=1; num[3][5]:=2; num[3][6]:=3; num[3][7]:=4; num[3][8]:=5; num[3][9]:=6;

num[4][1]:=2; num[4][2]:=3; num[4][3]:=4; num[4][4]:=5; num[4][5]:=6; num[4][6]:=7; num[4][7]:=8; num[4][8]:=9; num[4][9]:=1;

num[5][1]:=5; num[5][2]:=6; num[5][3]:=7; num[5][4]:=8; num[5][5]:=9; num[5][6]:=1; num[5][7]:=2; num[5][8]:=3; num[5][9]:=4;

num[6][1]:=8; num[6][2]:=9; num[6][3]:=1; num[6][4]:=2; num[6][5]:=3; num[6][6]:=4; num[6][7]:=5; num[6][8]:=6; num[6][9]:=7;

num[7][1]:=3; num[7][2]:=4; num[7][3]:=5; num[7][4]:=6; num[7][5]:=7; num[7][6]:=8; num[7][7]:=9; num[7][8]:=1; num[7][9]:=2;

num[8][1]:=6; num[8][2]:=7; num[8][3]:=8; num[8][4]:=9; num[8][5]:=1; num[8][6]:=2; num[8][7]:=3; num[8][8]:=4; num[8][9]:=5;

num[9][1]:=9; num[9][2]:=1; num[9][3]:=2; num[9][4]:=3; num[9][5]:=4; num[9][6]:=5; num[9][7]:=6; num[9][8]:=7; num[9][9]:=8;

**end**;

**procedure** change(alf:rec);//алфавитная замена

**var** i,j:integer;

**begin**

**for** i:=1 **to** 9 **do**

**for** j:=1 **to** 9 **do**

num[i][j]:=alf.num[(num[i][j])];

**end**;

**procedure** swip\_row();//обмен рядами

**var** i,j,k,r:integer;

**begin**

i:=random(1,9);

j:=i;

**while**(j=i) **do** j:=random(3\*((i-1+3)**div** 3-1)+1,3\*((i-1+3)**div** 3-1)+3);

**for** r:=1 **to** 9 **do**

**begin**

k:=num[i][r];

num[i][r]:=num[j][r];

num[j][r]:=k;

**end**;

**end**;

**procedure** swip\_line();//обмен столбцами

**var** i,j,k,r:integer;

**begin**

i:=random(1,9);

j:=i;

**while**(j=i) **do** j:=random(3\*((i-1+3)**div** 3-1)+1,3\*((i-1+3)**div** 3-1)+3);

**for** r:=1 **to** 9 **do**

**begin**

k:=num[r][i];

num[r][i]:=num[r][j];

num[r][j]:=k;

**end**;

**end**;

**procedure** wr();//вывод на экран

**var** i,j:integer;

**begin**

**for** i:=1 **to** 9 **do**

**begin**

**for** j:=1 **to** 9 **do**

write(num[i][j],' ');

writeln;

**end**;

writeln;

**end**;

**end**;

Подсчитав, получим что при замене только в одной тройке столбцов/рядов, можно увеличить количество вариантов в 6 раз на каждую тройку (6\*6\*6 за столбцы и столько же за строки, итого 46 656 вариантов).

А так же напишем отдельно функцию, которая будет посимвольно сравнивать две судоку между собой и возвращать true/false если они совпадают/не совпадают.

Код программы на Pascal

**function** compl(s1,s2:sudoku):boolean;//сравнение двух таблиц судоку

**var** i,j:integer; flag:boolean;

**begin**

flag:=true;

**for** i:=1 **to** 9 **do**

**for** j:=1 **to** 9 **do**

**if**(s1.num[i][j]<>s2.num[i][j])**then** flag:=false;

compl:=flag;

**end**;

**var**

a:rec;

nw,was:sudoku;

i:integer;

t,n:integer;

cor:integer;

**begin**

//Расчет вероятности совпадения судоку после замены алфавита и исходного варианта

cor:=0;

**for** t:=1 **to** 1000000 **do**

**begin**

nw:=**new** sudoku();

was:=**new** sudoku();

a:=**new** rec();

//Общее замечание. При нечетном количестве перестановок, исходная ситуация невозможна

//за 6 перестановок можно перемешать все тройки столбцов/строк

**for** i:=1 **to** 7 **do** nw.swip\_row();//6\*3 вариантов

**for** i:=1 **to** 7 **do** nw.swip\_line();//6\*3 вариантов

//за 8 парных перестановок можно получить любую комбинацию из 9 цифр.

**for** i:=1 **to** 9 **do** a.rand();//9! вариантов

//Оценочное количество вариантов 117 573 120

nw.change(a);

**if**(compl(nw,was)) **then** cor:=cor+1;

**end**;

writeln((cor/1000000)\*100,'%');

nw.wr();

**end**.

В заключении расчетов

Стоит пояснить, почему полученные два числа можно перемножить. При замене строк мы не получим тот же вариант, что и при замене цифр, так как иначе получим, другой алфавит замен (рассмотрев последнюю замену строк, можно найти противоречие).

Потому данный алгоритм позволяет создавать до 362 880 \* 46 656 или 16 930 529 280 вариантов.

В статье описаны не все преобразования с таблицей судоку (замена троек столбцов/строк, транспонирование и прочее), что доказывает, что количество вариантов судоку ещё больше.

\* ОПРОС \*

А как считаете вы, сколько вариантов судоку на самом деле?