

تمرین امتیازی بهینه سازی ترکیبیاتی محمد رضا باطنی 9812006 به نام خدا

مجموعه  $i$  و  $j$  نشان دهنده مختصات خانه و  $k$  مجموعه اعداد 1 تا 9 است و متغیر  $x_{i,j,k}$  نشان دهنده این است که آیا عدد  $k$  در خانه  $i, j$  قرار دارد یا خیر.

قید اول و دوم تضمین میکند که هر عدد در یک سطر و یک ستون فقط یک بار ظاهر شود.

قید سوم تضمین میکند که در هر خانه جدول فقط یک عدد ظاهر شود.

قید چهارم تضمین میکند خانه های جدول که مقدار دهی شدند، متغیر نظیرشان همان مقدار را نشان دهد.

قید پنجم تضمین میکند که در هر بلوک، هر عدد یک بار ظاهر شود.

در نهایت مدل توسط گمز حل میشود. به طوری که تابع `solve_with_gams` ورودی  $S$  را میگیرد که یک رشته 81 تایی از 0 ها و اعداد 1 تا 9 است که 0 نشان دهنده خالی بودن خانه متناظر و اعداد نشان دهنده پر بودن آن هاست.

در نهایت برنامه توسط کلاس `Interface` به صورت گرافیکی نمایش داده میشود و با کلیک کردن بر روی دکمه `solve` متغیر  $S$  ساخته میشود و به تابع داده میشود تا با استفاده از گمز حل شود. در نهایت خروجی  $S$  با استفاده از دو حلقه بر روی صفحه به نمایش در می آید.

نکته: برای اجرا شدن برنامه به هیچ پیشنهادی نیاز نیست و همه چیز اتوماتیک انجام میشود.

فقط کافیست گمز روی سیستم نصب باشد و تاریخ به عقب برگردانده شده باشد.

Simple Sudoku Game									
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	
			Solve		Clear				

Simple Sudoku Game									
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	
			Solve		Clear				

کد گمز برای وضوح بیشتر در زیر هم آمده است.

کد پایتون هم پیوست شده است.

**set**

i/1\*9/;

**alias**(i,j,k);

**Parameter**

n(i,j);

\$GDXIN %gdxincname%

\$LOAD n

\$GDXIN

**integer variable**

result,

v(i,j);

**variable**

z;

**binary variable**

*\*x[i,j,k] = 1 -> x(i,j) = k*

x(i,j,k);

**integer variable**

v(i,j);

**equation**

obj,

const1,

const2,

const3

const4,

const5;

obj..

z =e= 0;

*\*set variable with parameter*

const1(j,k) ..

**sum**(i,x(i,j,k)) =e= 1;

const2(i,k) ..

**sum**(j,x(i,j,k)) =e= 1;

const3(i,j) ..

```

    sum(k,x(i,j,k)) =e= 1;

const4(i,j,k)$(n(i,j)=k.val)..
    x(i,j,k) =e= 1;

*block
const5(i,j,k)$( (i.val = 1 or i.val = 4 or i.val = 7) and (j.val
= 1 or j.val = 4 or j.val = 7) )..

x(i,j,k)+x(i,j+1,k)+x(i,j+2,k)+x(i+1,j,k)+x(i+1,j+1,k)+x(i+1,j+2
,k)+x(i+2,j,k)+x(i+2,j+1,k)+x(i+2,j+2,k) =e= 1;

model sudoku/obj, const1, const2, const3, const4, const5/;
solve sudoku using MIP minimizing z;

```