What is Sudoku ? 

Number-placement puzzle which objective is to fill the 9x9 grid.

So make the input :

question = np.array(

[[5, 3, 0, 0, 7, 0, 0, 0, 2],

[6, 0, 0, 1, 9, 5, 0, 0, 8],

[0, 9, 8, 0, 0, 0, 0, 6, 7],

[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]

]

Each row,

row\_C = np.append(np.ones(1),np.zeros(N-1))

row\_R = np.append(np.ones(1),np.zeros(N-1))

row\_R = np.reshape(row\_R,[N,1])

row = toeplitz(row\_C,row\_R)

ROW = np.kron(row,np.kron(np.ones(9),np.eye(N)))

ROW = sparse.csc\_matrix(ROW)

Each column,

col\_R =np.kron(np.ones(9),np.append(np.ones(1),np.zeros(N-1)))

col = toeplitz(row\_C,col\_R)

COL = sparse.csc\_matrix(np.kron(col, np.eye(N)))

and each of the nine 3x3 subgrids

box\_R = np.kron(np.ones(sqrtN),np.append(np.ones(1),tep))

box\_C = np.append(np.ones(1),np.zeros(sqrtN-1))

box\_R = np.reshape(box\_R,[N,1])

box = toeplitz(box\_C,box\_R)

box2 = np.kron(np.eye(sqrtN),box)

eye3N = np.append(np.eye(N),np.eye(N),axis = 1)

eye3N = np.append(eye3N,np.eye(N),axis = 1)

BOX = sparse.csc\_matrix(np.kron(box2,eye3N))

that compose the grid contain all of the digits from 1 to 9.

Transfer the Sudoku to math problem

There have 81 block \* 9 number = 729 unknowns

Our object and constraint can be transfer to

def sudoku\_obj(x):

return sum(x)-81

def constraint1(x):

return A\*x-1

Solve the minimize by sequential least squares programming

con1 = {'type': 'ineq', 'fun': constraint1}

solution = minimize(sudoku\_obj,x0,method='SLSQP',bounds=bnds,constraints=con1)

Solution:

[[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]]

Tool : **pycharm** or **colab**

Lib : **numpy** , **scipy**

Copy and Paste **main.py** code to colab or

Execution **main.py in python** environment like pycharm