Sudoku solving

In [89]:

```
import numpy as np
```

In [90]:

```
grid=[[7,0,0,0,0,5,0,0,0],[0,5,0,0,0,4,0,6,9],[3,0,0,0,0,6,0,2,0],[0,0,0,0,5,0,9,0,8],
      [5,0,2,0,9,0,0,0,0],[0,4,0,2,0,0,0,0,5],[6,3,0,4,0,0,0,8,0],[0,0,0,5,0,0,0,0,6]]
```

In [91]:

```
print(np.matrix(grid))
```

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

In [92]:

```
def possible(y,x,n):
    global grid
    for i in range(0,9):
        if grid[y][i]==n:
            return False
    for i in range(0,9):
        if grid[i][x]==n:
            return False
    x0=(x//3)*3
    y0=(y//3)*3
    for i in range(0,3):
        for j in range(0,3):
            if grid[y0+i][x0+j]==n:
                return False
    return True
def solve():
    global grid
    for y in range(9):
        for x in range(9):
            if grid[y][x]==0:
                for n in range(1,10):
                    if possible(y,x,n):
                        grid[y][x]=n
                        solve()
                        grid[y][x]=0
                return
    print(np.matrix(grid))
```

In [93]:

```
1 solve()
```

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

In [ ]:

```
1
```