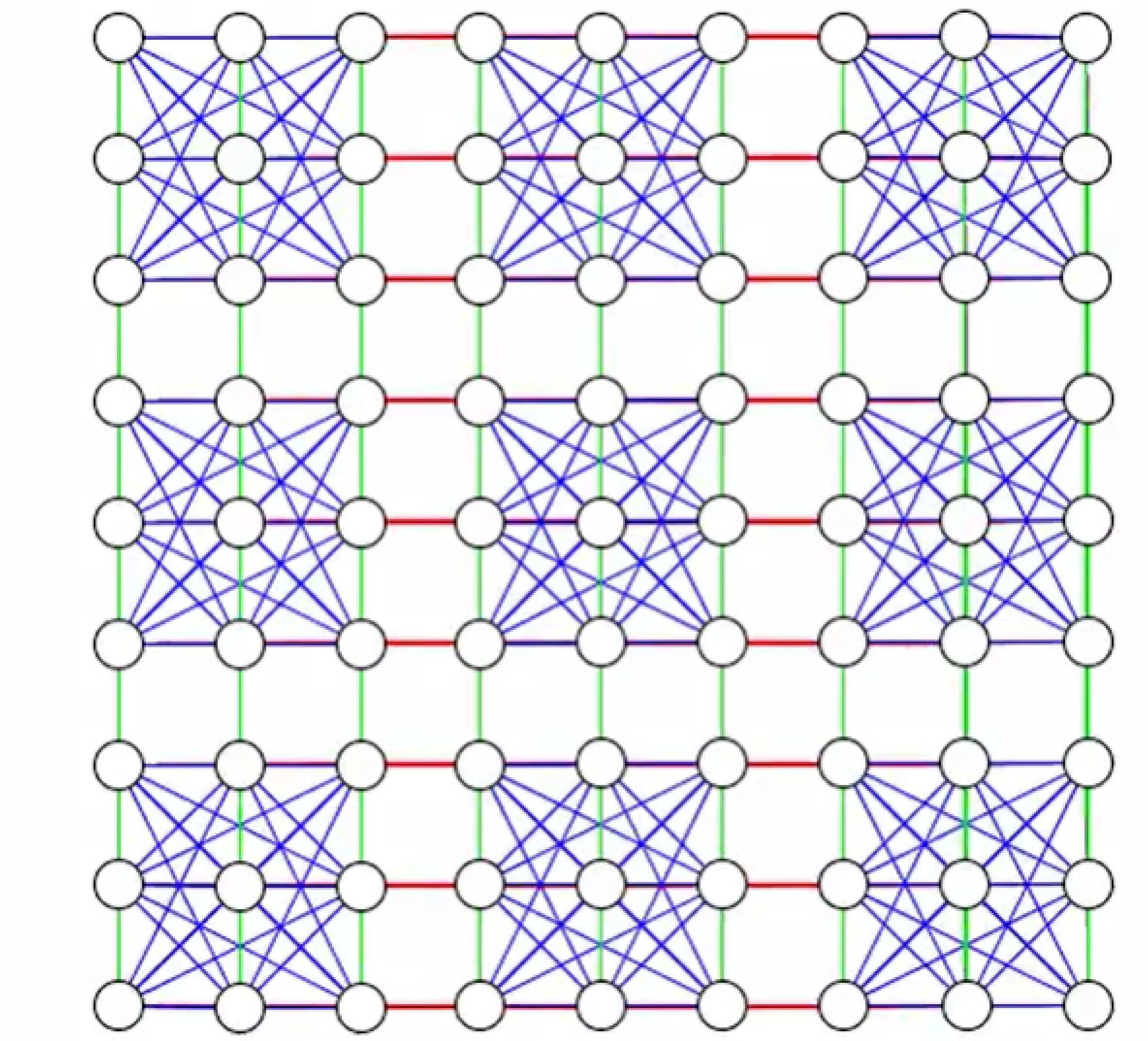
Visualization of the problem:
Initially for solving the problem,we have to check for a better description of the sudoku grid.Instead of simply observing that as a grid,we have to change our convention for simplifying the problem. For representing we have two constraints:
1.Each element in the sub grid 3*3 is interlinked, that is they should be distinct with each other for solving the puzzle.
2.Similarly the row and columns of the main grid are interlinked since they should also have connectivity between the nodes for the correct solution.

This makes the simplified solution as:



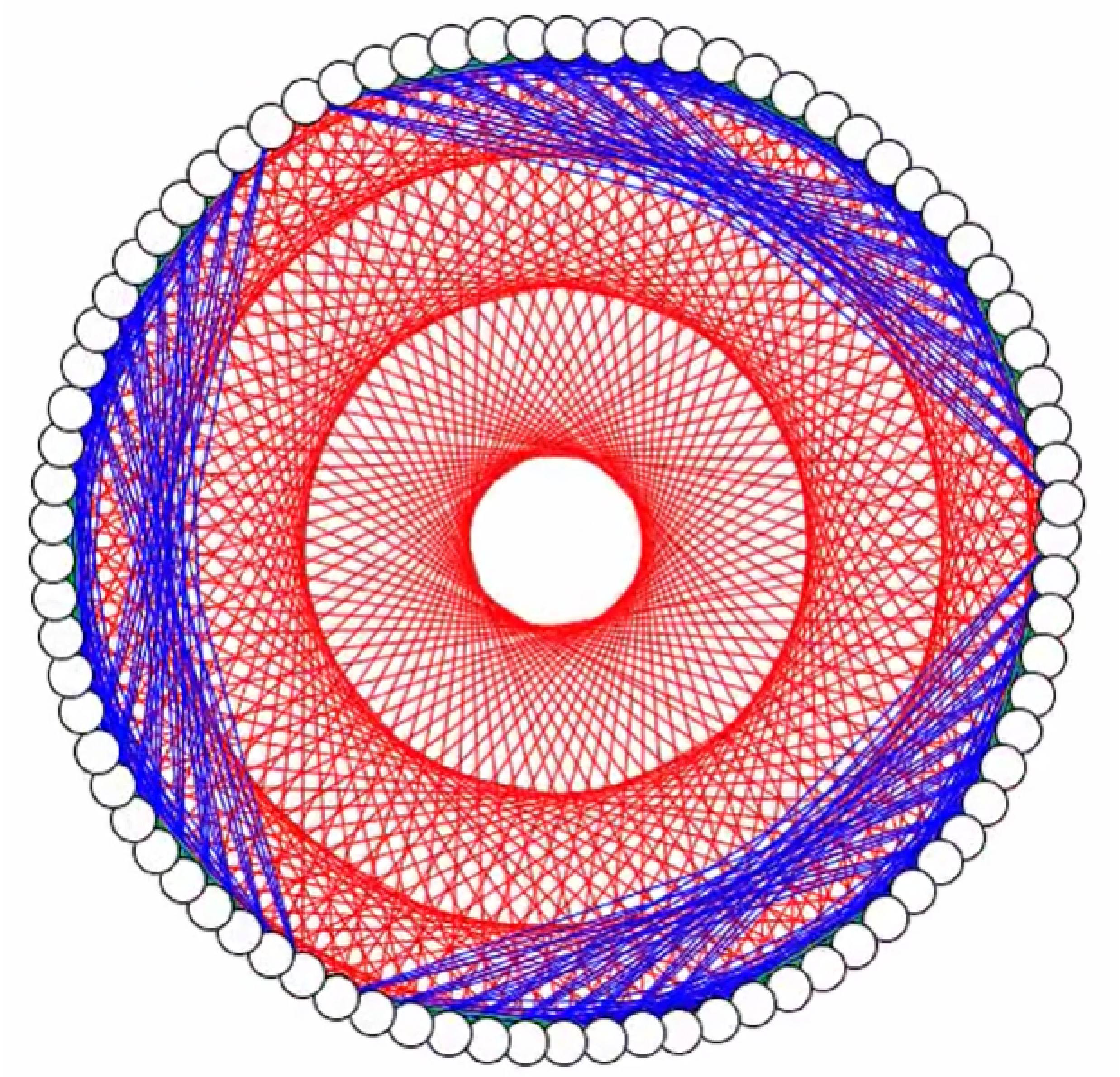
In this figure,
All edges are not clearly shown due to 2D constraint hence,the interlinking of the elements in the 1.same row/column

2.Overlapping edges are not shown.

Analyzing the figure:

•The circles represent any digit 1-9 each
•The green lines connect circles that can't be the same digit because they're in the same column
•The red lines connect circles that can't be the same digit because they're in the same row
•The blue lines connect circles that can't be the same digit because they're in the same 3x3 square.
•Green and red are most prominent.Only 4 blue lines from any single circle are visible because the other blue lines are occluded from either green or red since those circles are also in this same row or column in addition to being in the same 3x3 square.

We can change this to into a more simplified image which looks as



We can observe the exact connectivity between each node clearly.
We can view this problem as a graph in which we have to colour such that no adjacent will have same colour.There are 9 distinct elements:1-9 hence we need 9 colors for satisfying this constraint.Hence,the chromatic number=9.

Research:
I have taken this idea of visualizing from:
1.BS-MS Thesis: Graph Theory of Sudoku
2.Reddit:https://www.reddit.com/r/dataisbeautiful/comments/6ty4vf/visualizing_the_sudoku_connectivity_graph_more_in/

Pseudo code initialization:

We have divided the total code portion into several individual small segments which are:

1.Creating a Node and a graph:

We have to make a node class which has initialization,inserting functions.

We have to make a graph class next with which we can be able to connect all nodes and there should be a function for adding an edge to a graph with an interlink between nodes. *check whether the initial graph segment is working without errors or not else errors will pile up and make difficult for detection.

2.Making connections as in grid:

First import the graph from made previously.We make a separate connections file which have its initialization and connections as in the grid.Then,we finally generate actual graph

3.Better representing and 3*3 grid connections:

We have to arrange the nodes in a matrix using getgridmatrix() which will arrange all the nodes in a better representation.Next we have to make sure to add constraints needed for sudoku for which we have to add 3 connections:

1.Connecting Each Node to every other node in the same row
2.Connecting Each Node to every other node in the same column
3.Connecting Each Node to every other node in the same sub grid(3*3 block)

We merge all of these into a main function with which we can generate a graph with all required constraints from a 2D array as an input.
All left now is to make coloring for the graph obtained so that we will get the solution for sudoku

5.Graph coloring algorithm

With this all coding portion will be completed.

Input:2D array of numbers with 0 as blanks
Output:solved sudoku grid with good representation