

1. Super Graph Coloring

Greedy Graph Coloring algorithm is implemented in Java.

The Greedy approach works by checking if there is a color assigned to each node.

- 1) Consider the currently picked vertex and color it with the lowest-numbered color that has not been used on any previously colored vertices adjacent to it.
- 2) For any unavailable color, Add a new color.
- 3) Once color is assigned to a node, then it considers the next neighbor vertex.

```
praveenakondepudi@Praveenas-MacBook-Pro → src git:(main) ✕ javac project/SuperGraphColoring.java
Note: project/SuperGraphColoring.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
praveenakondepudi@Praveenas-MacBook-Pro → src git:(main) ✕ java project.SuperGraphColoring
Vertices: 9 Edges: 18
Maximum number of edges : 36
Total colors : 3
Addl. new colors used : 1
1 2 3 2 3 1 3 1 2 3
praveenakondepudi@Praveenas-MacBook-Pro → src git:(main) ✕
```

2. Number of colors (Random Graph) & Maximum degree

Vertices	Edges	Colors
50	100	4
100	1000	9
200	4000	14
300	10000	20
400	20000	28
500	50000	50

Vertices	Max Edges	Colors
50	1225	50
100	4950	100
200	19900	200
300	44850	300
400	79800	400
500	124790	500

The maximum number of edges in an undirected graph is $n(n-1)/2$, the maximum number of colors are equal to number of vertices for maximum number of edges.

3. Sudoku Graph

For 81 vertices and 810 edges below is output

```
praveenakondepudi@Praveenas-MacBook-Pro → src git:(main) ✕ java project/SudokuSolver.java
<-----Input Sudoku----->
-1,-1,1,-1,-1,-1,-1,-1,4,
-1,2,3,1,-1,-1,9,-1,-1,
-1,8,6,-1,5,-1,3,-1,-1,
-1,6,-1,-1,4,-1,-1,-1,-1,
3,-1,-1,9,-1,1,-1,-1,5,
-1,-1,-1,-1,7,-1,-1,3,-1,
-1,-1,4,-1,3,-1,6,1,-1,
-1,-1,2,-1,-1,5,8,4,-1,
7,-1,-1,-1,-1,-1,5,-1,-1,
<-----Output Sudoku----->
Maximum colors available: 9
5,7,1,3,9,6,2,8,4,
4,2,3,1,8,7,9,5,6,
9,8,6,4,5,2,3,7,1,
2,6,7,5,4,3,1,9,8,
3,4,8,9,2,1,7,6,5,
1,9,5,6,7,8,4,3,2,
8,5,4,2,3,9,6,1,7,
6,3,2,7,1,5,8,4,9,
7,1,9,8,6,4,5,2,3,
praveenakondepudi@Praveenas-MacBook-Pro → src git:(main) ✕
```