**Graph coloring algorithm**

Graph coloring is the procedure of assignment of colors to each vertex of a graph G such that no adjacent vertices get same color. The objective is to minimize the number of colors while coloring a graph. The smallest number of colors required to color a graph G is called its chromatic number of that graph. Graph coloring problem is a NP Complete problem.

**Method to Color a Graph**

The steps required to color a graph G with n number of vertices are as follows −

**Step 1** − Arrange the vertices of the graph in some order.

**Step 2** − Choose the first vertex and color it with the first color.

**Step 3** − Choose the next vertex and color it with the lowest numbered color that has not been colored on any vertices adjacent to it. If all the adjacent vertices are colored with this color, assign a new color to it. Repeat this step until all the vertices are colored.

**Input:**

graph = {0, 1, 1, 1},

{1, 0, 1, 0},

{1, 1, 0, 1},

{1, 0, 1, 0}

**Output:**

Solution Exists:

Following are the assigned colors

1 2 3 2

**Explanation:** By coloring the vertices with following colors, adjacent vertices does not have same colors

**Analysis:**

**Time Complexity:**  O(m^V).   
There are total O(m^V) combination of colors. So time complexity is O(m^V). The upperbound time complexity remains the same but the average time taken will be less.

**Space Complexity:**  O(V).   
Recursive Stack of graphColoring(…) function will require O(V) space.