Graph Coloring

VERTEX COLORING OF GRAPHS

- A graph is said to be k vertex colorable (or k-colorable) if it is possible to assign one color from a set of k colors to each vertex such that no two adjacent vertices have the same color
- If the graph G is k-colorable but not (k - 1) colorable, can say that G is a k-chromatic graph and that its **chromatic number X(G) = k**
- the **chromatic number is the minimum number k** such that G is k-colorable
- Hence, graph G is k-colorable iff X(G)<= k

VERTEX COLORING OF GRAPHS

- A graph is said to be k vertex colorable (or k-colorable) if it is possible to assign one color from a set of k colors to each vertex such that no two adjacent vertices have the same color
- If the graph G is k-colorable but not (k - 1) colorable, can say that G is a k-chromatic graph and that its **chromatic number X(G) = k**
- the **chromatic number is the minimum number k** such that G is k-colorable
- Hence, graph G is k-colorable iff X(G)<= k

➔k-chromatic graph → graph needs at least k colors (Lower Bound)

➔k colorable graph → graph that does not need more than k colors (Upper Bound)

Observations

- X(G) = 1 iff and only if G is trivial
- X(G) = 2 iff G is bipartite
- X(G) = 3 → $_{\text{e.g.}}$ i) any cycle with an odd number of vertices

  ii) wheel of even order
- X(G) <= max_degree(G) + 1   //in general//
- X(G) = max_degree(G) + 1

//G is a complete graph or an odd cycle//

- X(G) = n → $_{\text{e.g}}$ The complete graph K_n
- X(G) = max_degree(G)

//G is a connected; & is neither a complete graph nor an odd cycle //

- X(G)$_{\text{Upper bound}}$ can be achieved by adopting a **Greedy method** that is also known as a **Sequential (incremental) coloring algorithm**

EDGE COLORING OF GRAPHS

- A graph G with no loops is said to be k edge colorable if it is possible to assign to each edge one color from a set of k colors such that no two edges with a vertex in common get the same color
- A k edge colorable graph is a k edge chromatic graph if it is not (k - 1) edge colorable and if its chromatic index $X'(G) = k$
- chromatic index of a simple graph G = chromatic number of its line graph L(G)

→ since two edges in G have a vertex in common iff the vertices corresponding to these edges are adjacent in L(G)

→ chromatic number = chromatic index for any cyclic graph

Graph Coloring Algorithm1

- No efficient graph coloring algorithm for a with minimum number of colors; Graph Coloring is a **NP complete** problem
- However, **Greedy algorithm** is known for finding the chromatic number of any given graph

Step-01: Color first vertex with the first color

Step-02: for remaining (V-1) vertices one by one and do the following-

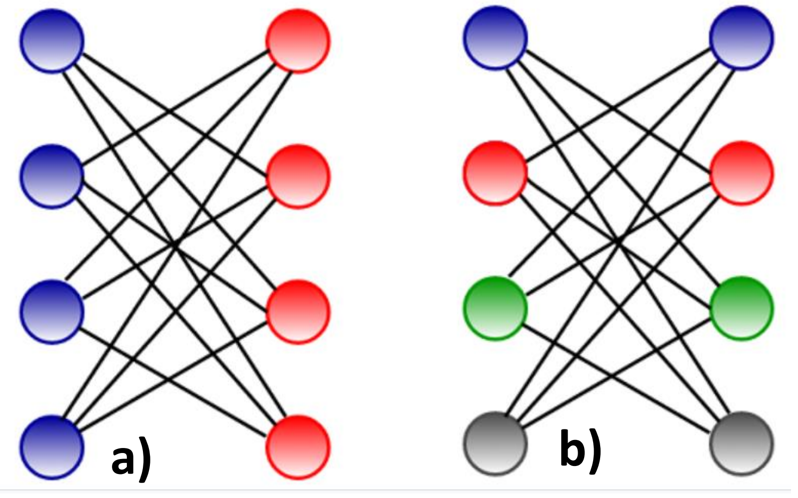
- Color the currently picked vertex with the lowest numbered color if it has not been used to color any of its adjacent vertices

- If it has been used, then choose the next least numbered color

- If all the previously used colors have been used, then assign a new color to the currently picked vertex

Drawbacks of Greedy Algorithm:

- The Greedy algorithm does not always use minimum number of colors
- The number of colors used sometimes depend on the order in which the vertices are processed



- For a Graph with maximum degree of x; Greedy algorithm uses maximum (x+1) colors
- a) uses 2 colors only; where Greedy algorithm in b) uses n/2 = 4 colors

Graph Coloring Algorithm2

Naive Algorithm:

- This approach uses the **brute force method**
- Finds all permutations of color combinations that can color the graph
- If any of the permutations is **valid** for the given graph and colors, output the result otherwise not
- **Not efficient** in terms of **time complexity** because it finds all colors combinations rather than a single solution

Complexity =???

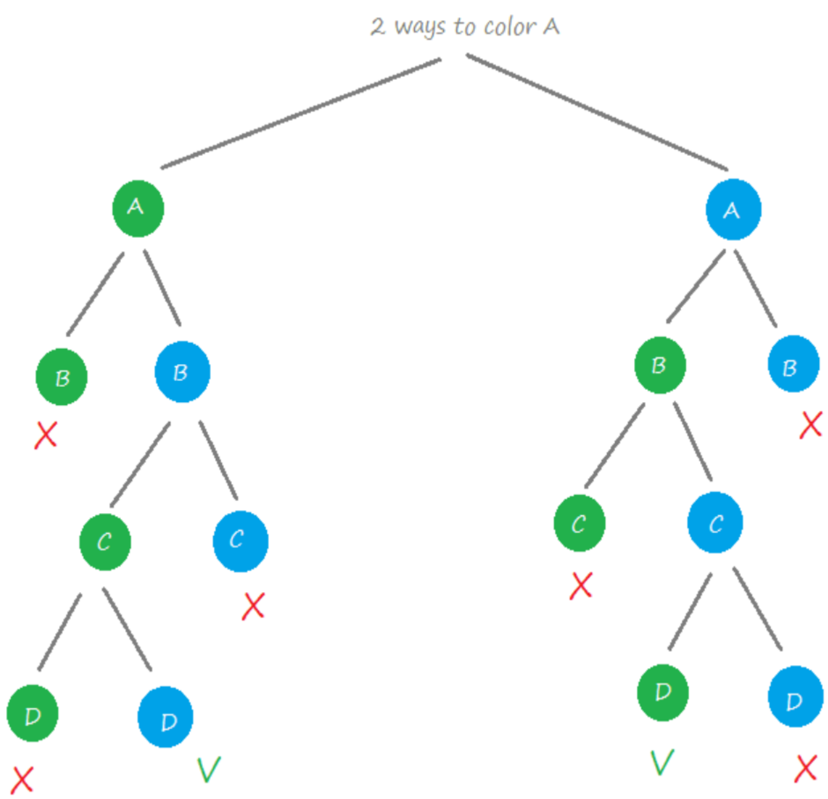
Naive Algorithm: Complexity

- Since each node (No. of nodes = v) can be colored by using any of the m colors, the total number of possible color configurations are m^V
- The complexity is **exponential** which is very huge
- Time Complexity: **$O(m^V)$**
- Space Complexity: **$O(V)$** which is for storing the output array of nodes

Graph Coloring Algorithm3

Backtracking Algorithm:

- Efficient as compared to Naïve algorithm



Task 1: Continue *– try a different color for current vertex looking to the adjacent vertex color*

Task 2: Backtrack *– try a different color for last colored vertex*
(i.e. un-color last colored vertex)

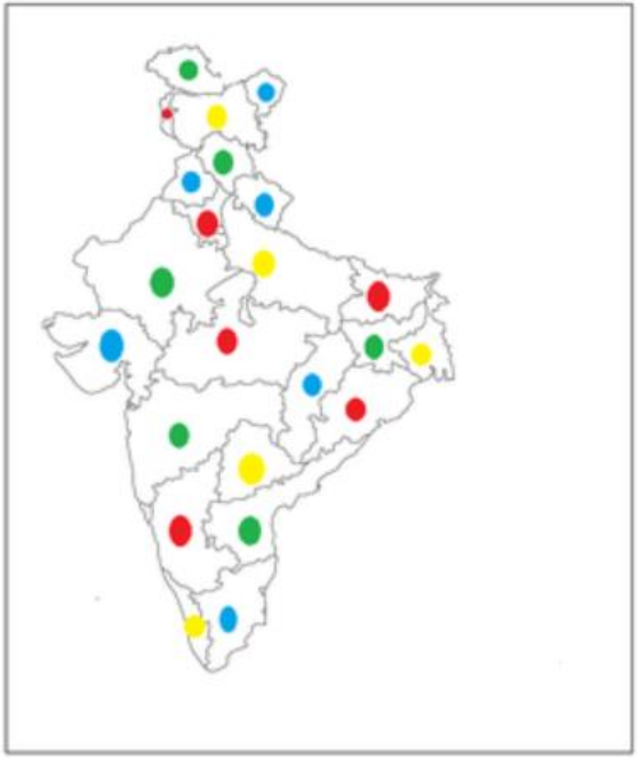
Note: *backtrack arrives to the last recursive call to change the color of the last colored vertex.* ***If false is returned by the root → no solution*** *for given graph coloring problem*

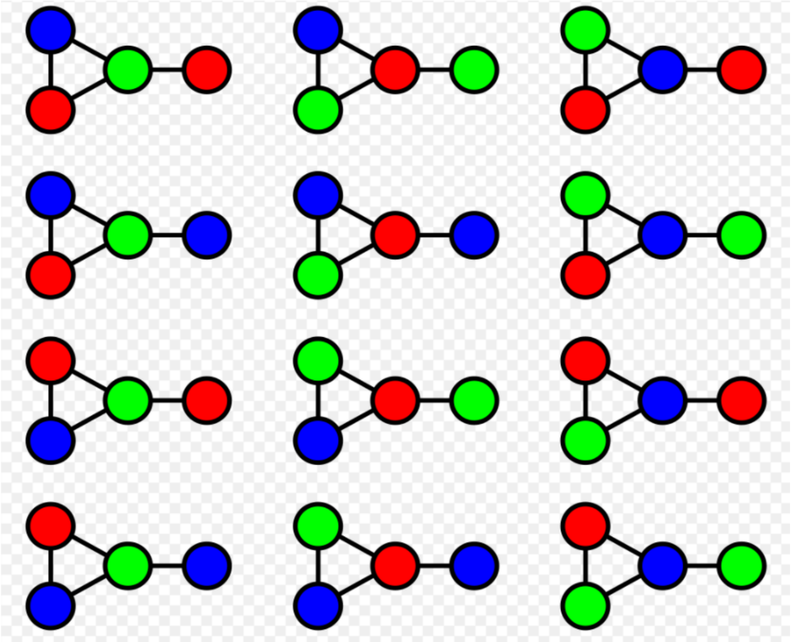
Backtracking Algorithm: Complexity

- **Time Complexity**: **$O(m^V)$**
- Since backtracking is **also a kind of brute force** approach, there would be total $O(m^V)$ possible color combinations
- It is to be noted that the upper bound time complexity remains the same but the **average time** taken will be **less** due to the refined approach
- **Space Complexity**: **$O(V)$** for storing the output array of nodes

Every planar graph is 5-colorable - Prof by Induction

- **Every planar graph is 4-colorable** (Vertex Coloring)
- - but when a triangle is a graph or sub-graph we need only 3 colors.





This graph can be 3-colored in 12 different ways

Graph Colors_ Theorems

Euler's Formula

Euler's formula for polyhedra (and, hence for connected planar graphs …):

$$v - e + f = 2$$

(where v is the number of vertices, e is the number of edges, and f is the number of faces)

- A graph is collection of vertices, some of which are connected by edges
- A face is a region surrounded by edges, with no vertices or edges in the interior
- There is one large exterior face surrounding everything and going off to infinity in all directions

Corollary_ Euler formula:

- Note: each face is bounded by at least 3 edges, and each edge is part of the boundary of a face twice (once on each side of the edge), Hence:

$$3f \leq 2e$$

Now, from Euler's formula, we have:

$$f = 2 - v + e$$

➔

$$3(2 - v + e) \leq 2e$$

OR

$$\mathbf{2 - v + e \leq (2/3) * e \quad \ldots (A)}$$

Continuing Further as:

$$2 - v + e \leq (2/3)*e \quad \ldots (A)$$

$$\rightarrow (1/3)*e <= v - 2$$

$$\rightarrow e <= 3v - 6 \quad \ldots (B)$$

- Sum of all the degrees of the vertices is equal to twice the number of edges.
- If all the degrees are greater than or equal to say 6; then

$$6v \leq sum\ of\ degrees = 2e$$

OR $3v \leq e$ $\rightarrow$ *contradicts with B*

OR $3v \leq e \leq 3v - 6$ $\rightarrow$ *contradicts with B*

➔Every planar graph has a vertex of degree 5 or fewer

≈ there is at least one vertex with degree 5 or fewer

The 6-Color Theorem

Base case: The simplest connected planar graph consists of a single vertex. Pick a color for that vertex.

Induction step: Assume $k \geq 1$, and assume that every planar graph with k or fewer vertices can be 6-colored.

Proof: Consider a planar graph with $k + 1$ vertices.

However, the graph has a vertex of degree 5 or fewer.

-Remove that vertex (and all edges connected to it)

-By the induction hypothesis, we can <u>6-color the remaining graph</u>.

-Put the vertex (and edges) back in

-We have a graph with every vertex colored (without conflicts) except for the "special" one

-There are at most 5 colors adjacent, so we have at least one color left. Use an available color for that vertex

→ 6-colored the graph

The 5-Color Theorem

Theorem 2. *Every planar graph is 5-colorable*

Base case: The simplest connected planar graph consists of a single vertex. Pick a color for that vertex.

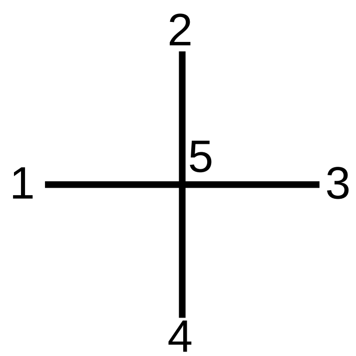
Induction step: Assume $k \geq 1$, and assume that every planar graph with k or fewer vertices can be 5-colored. Now consider a planar graph with $k + 1$ vertices.

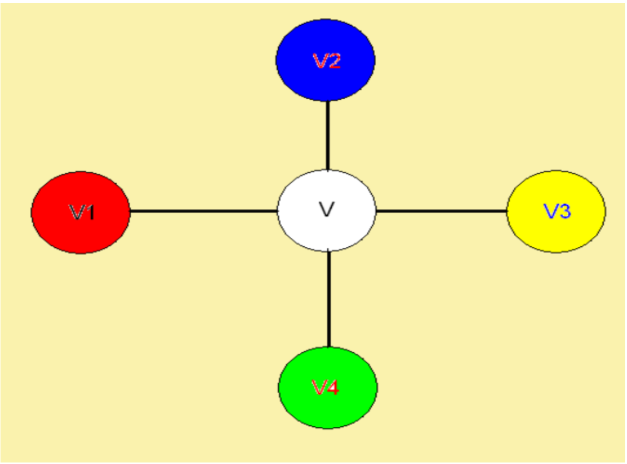
the graph has a vertex of degree 5 or fewer. Remove that vertex (and all edges connected to it). By the induction hypothesis, we can <u>5-color the remaining graph</u>. Put the vertex (and edges) back in. We have a graph with every vertex colored (without conflicts) except for the one.

Case1: deg (v) <=4 (i.e. If the vertex has degree less than 5)

There are at most 4 colors that have been used on the neighbors of v. There is at least one color then available for v.

→ G can be colored with five colors

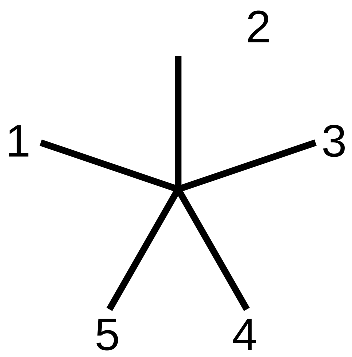




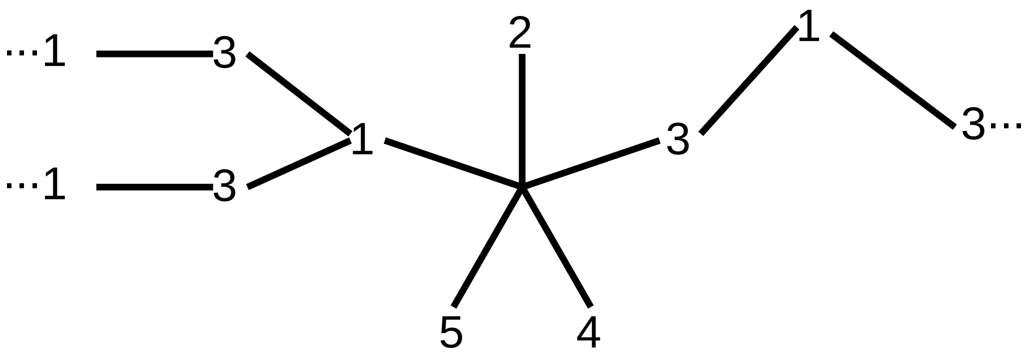
Case 2: deg (v)= 5

If the vertex has degree 5, and all 5 colors are connected to it.

In this case, using numbers 1 through 5 to represent colors, label the vertices adjacent to the “special” (degree 5) vertex 1 through 5 (inorder).

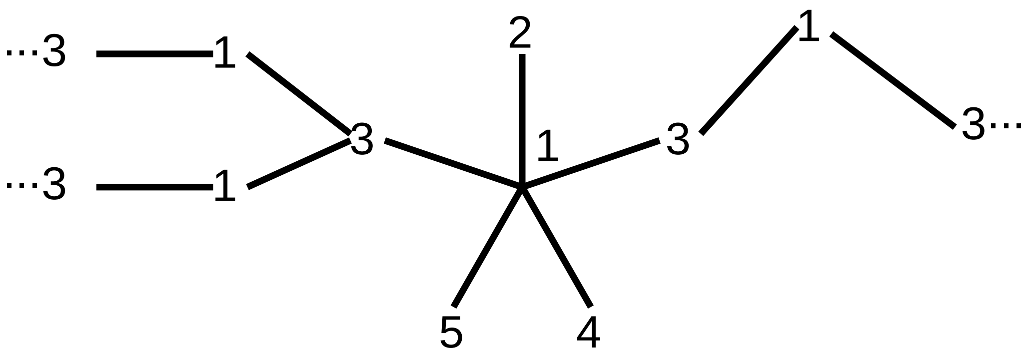


Now make a subgraph out of all the vertices colored 1 or 3 which are connected to the 1 and 3 colored vertices adjacent to the ”special” vertex.

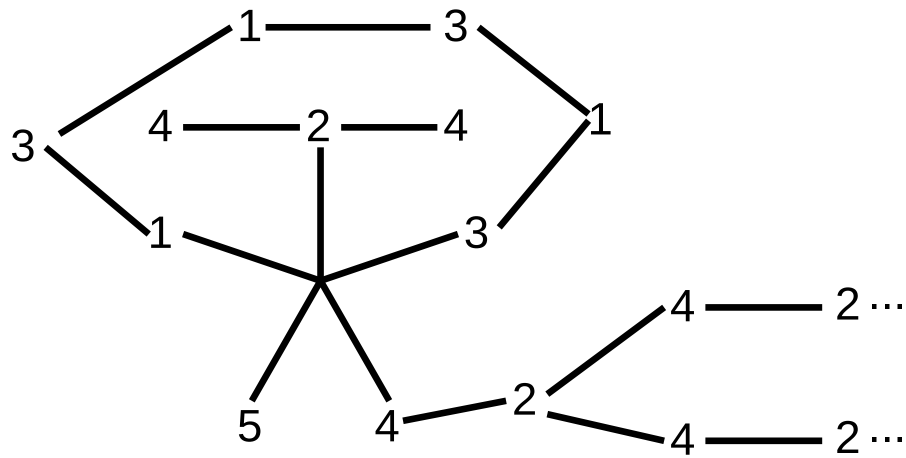


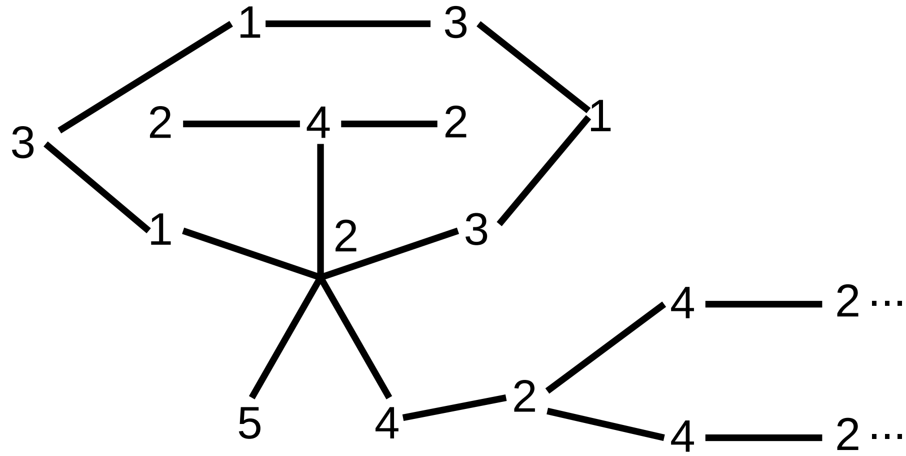
If the adjacent vertex colored <u>1 and the adjacent vertex colored 3 are not connected</u> by a path in this subgraph, simply exchange the colors 1 and 3 throughout the subgraph connected to the vertex colored 1.

This will leave color 1 available to color the ”special” vertex.



On the other hand, if the <u>vertices colored 1 and 3 are connected</u> via a path in the subgraph, this will be a disconnected pair of subgraphs, separated by a path connecting the vertices colored 1 and 3. Now we can exchange the colors 2 and 4 in the subgraph connected to the adjacent vertex labeled 2. This will leave color 2 for the ”special” vertex.

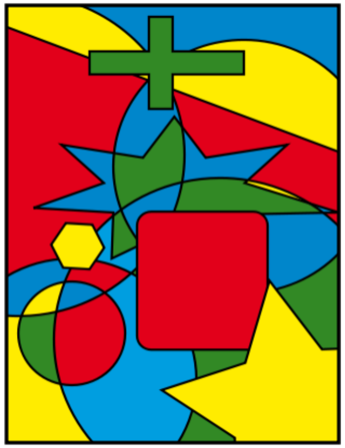


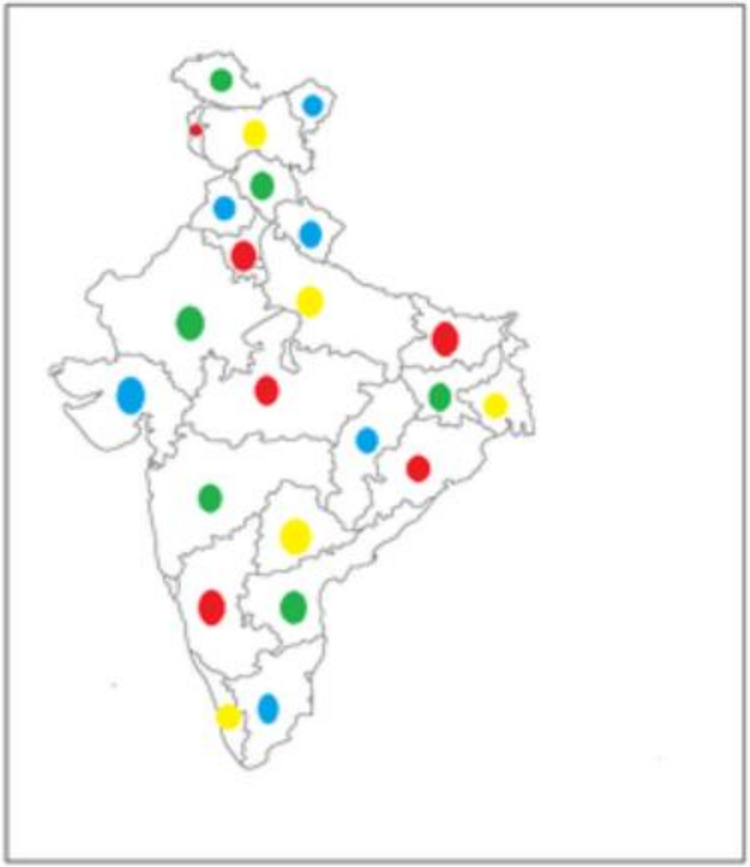


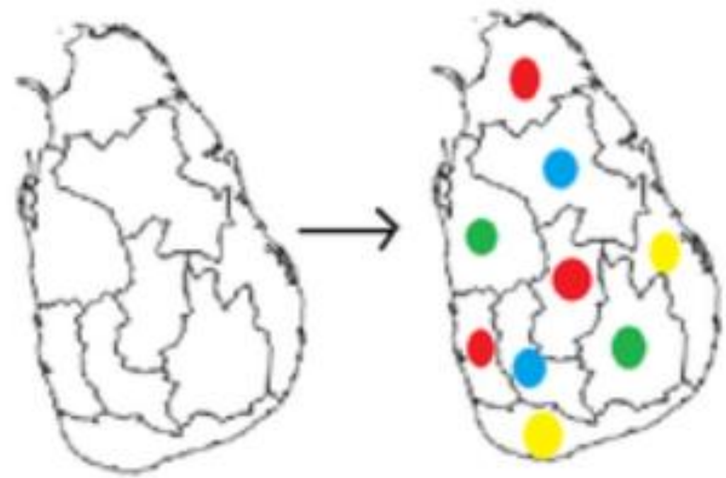
Thus, we will be able to color the entire planar graph with 5 colors

The 4-Color Theorem

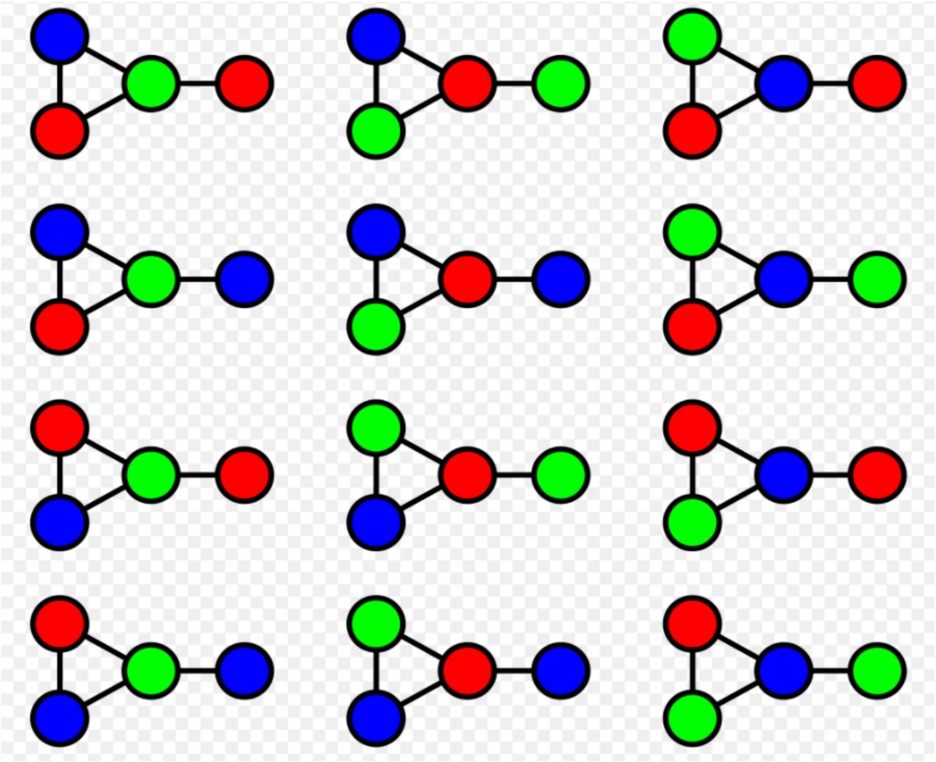
The four color theorem, or the four color map theorem, states **that given any separation of the plane into contiguous regions, called a "map"**, the regions can be colored using at most four colors so that no two adjacent regions have the same color.







Every planar graph is 4-colorable (Vertex Coloring) but when a triangle is a graph or sub-graph we need only 3 colors.



This graph can be 3-colored in 12 different ways

- https://www.interviewbit.com/tutorial/graph-coloring-algorithm-using-backtracking/
- https://www.geeksforgeeks.org/m-coloring-problem-backtracking-5/