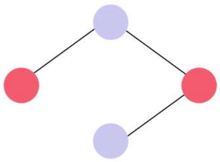
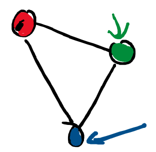
S

13 April 2025 12:25

We are given a graph, we need to assign colours to the vertices of the graph. In the graph colouring problem, we have a graph and m colours, we need to find a way to colour the vertices of the graph using the m colours such that any two adjacent vertices are not having the same colour. The chromatic number is the minimum number of colors needed to color the graph with the constraint that no two adjacent vertices have the same color.





Color = {R, G, B}

🔹 Backtracking Approach — Theory

Backtracking is a **depth-first search**-based technique that **builds the solution step-by-step**, and **backtracks** as soon as a constraint is violated.

1. State Space

Each state represents a **partial assignment** of colors to the graph's vertices.

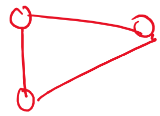
2. Decision Tree

At each level of the recursion:

- Pick a vertex.
- Try assigning each of the **M colors** to that vertex.
- Check if this color is **safe** (i.e., none of its adjacent vertices has the same color).
- If yes, **recurse** for the next vertex.
- If no color works, **backtrack**.

3. Base Case

If all vertices are colored without conflict → **Solution found**.



m = {R, G}

Not possible to color

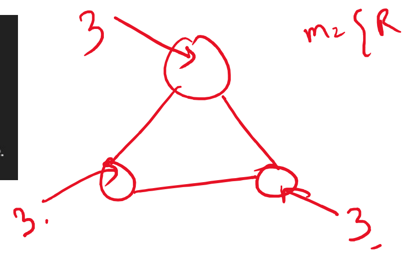
🔹 Time Complexity

In the worst case:

O(M^V) — M choices (colors) for each of V vertices.

But backtracking **prunes** invalid paths early, making it faster in practice.

M^V

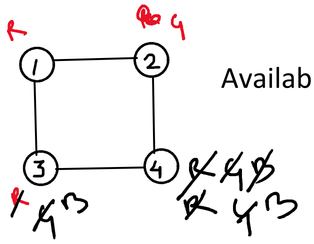


m = {R, G, B}.

TC → $O(M^V)$ 3^4

Possible comb^n = 3 × 3 × 3 × 3

= 3^4 = 81.



Available colours= {R, G, B}

Solution

1	2	3	4
R	G	G	R
R	G	G	B
R	G	B	R