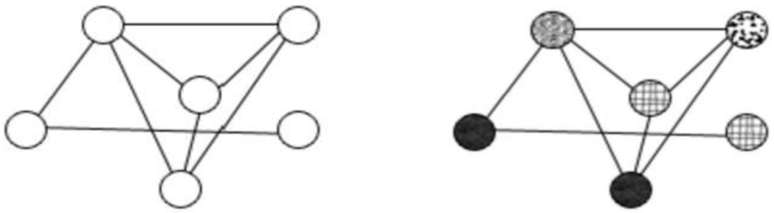
CSE 245: Algorithm

Project: Graph Coloring Problem

Graph Colouring problem is to assign colours to certain elements of a graph subject to certain constraints. The problem is, given an undirected graph and a number m, determine if the graph can be coloured with at most m colours such that no two adjacent vertices of the graph are coloured with a same colour. Here colouring of a graph means assignment of colours to all vertices. For example,



Non-coloured **Coloured**

The idea is to assign colours one by one to different vertices, starting from the vertex 0. Before assigning a colour, check for safety by considering already assigned colours to the adjacent vertices.

In this project, you have to design a suitable algorithm that solves the Graph Colouring problem generating all possible combinations of colours with the given constraints. You should use a **Greedy** approach or a recursive algorithm based on **Backtracking** approach in this regard as they have the ability to give same result in far fewer attempts than Brute Force method trials and thus handles the complexity better

Input:

1) A 2D array graph[V][V] where V is the number of vertices in graph and graph[V][V] is adjacency matrix representation of the graph. A value graph[i][j] is 1 if there is a direct edge from i to j, otherwise graph[i][j] is 0.
2) An integer m which is maximum number of colors that can be used.

Output:

An array color[V] that should have numbers from 1 to m. Array color[i] should represent the color assigned to the ith vertex. The code should also return false if the graph cannot be colored with m colors.