

M-Coloring Problem

Given an undirected graph and an integer **M**. The task is to determine if the graph can be colored with at most M colors such that no two adjacent vertices of the graph are colored with the same color. Here coloring of a graph means the assignment of colors to all vertices. Print 1 if it is possible to colour vertices and 0 otherwise.

Example 1:

Input:

N = 4

M = 3

E = 5

Edges[] = { (0,1), (1,2), (2,3), (3,0), (0,2) }

Output: YES

Explanation: It is possible to colour the given graph using 3 colours.

Example 2:

Input:

N = 3

M = 2

E = 3

Edges[] = { (0,1), (1,2), (0,2) }

Output: NO

Your Task:

Your task is to complete the function **graphColoring()** which takes the 2d-array graph[], the number of colours and the number of nodes as inputs and returns “**YES**” if answer exists otherwise “**NO**”. 1 is printed if the returned value is **true**, 0 otherwise. The printing is done by the driver's code.

Note: In Example there are Edges not the graph.Graph will be like, if there is an edge

between vertex X and vertex Y graph[] will contain 1 at graph[X-1][Y-1], else 0. In 2d-array graph[], nodes are 0-based indexed, i.e. from 0 to N-1. Function will be contain 2-D graph not the edges.

Expected Time Complexity: $O(M^N)$.

Expected Auxiliary Space: $O(N)$.

Constraints:

$$1 \leq N \leq 20$$

$$1 \leq E \leq (N*(N-1))/2$$

$$1 \leq M \leq N$$