

16. M-Coloring Problem

M-Coloring Problem

Medium Accuracy: 47.46% Submissions: 42636
Points: 4

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.

Input:

$N = 4$

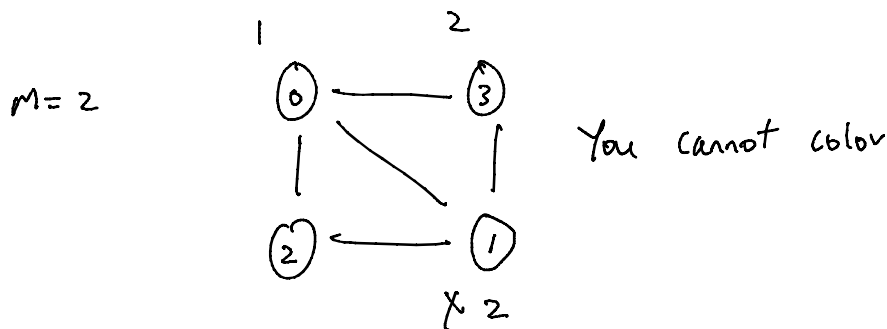
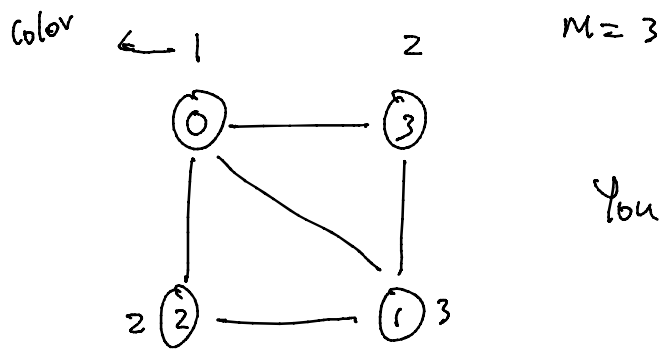
$M = 3$

$E = 5$

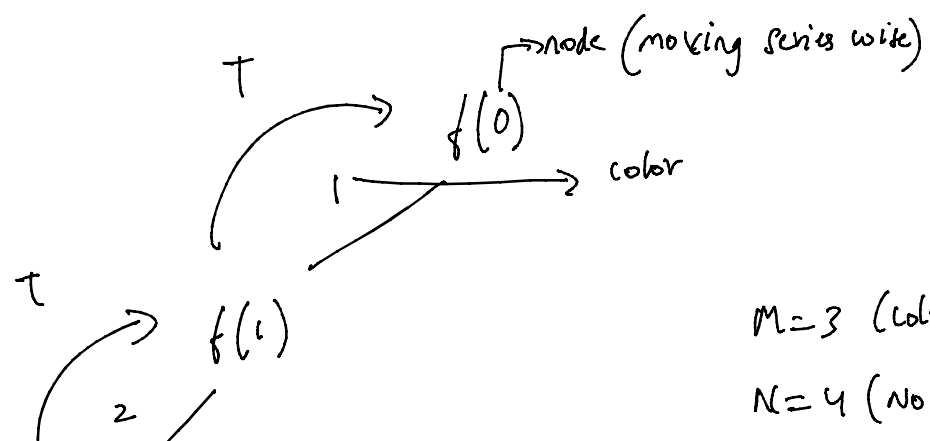
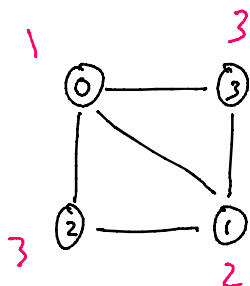
$\text{Edges[]} = \{(0,1), (1,2), (2,3), (3,0), (0,2)\}$

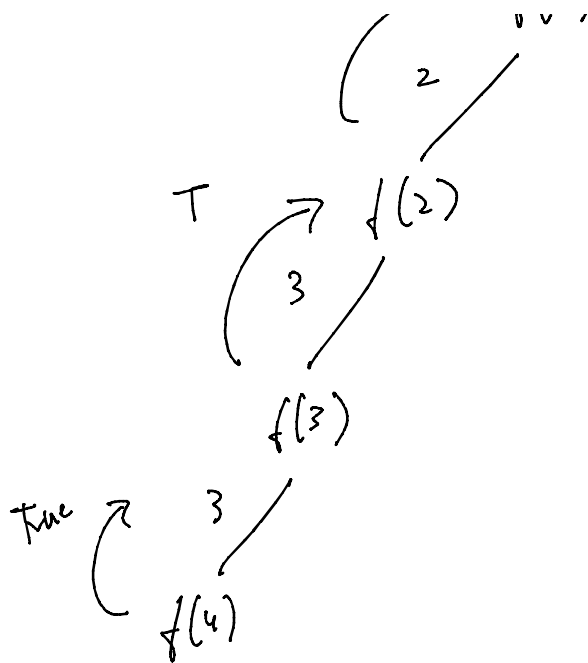
Output: 1

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



$M = 3$





$N=4$ (Node)

```

f(node)
{
    // base case
    if (node == n) return true

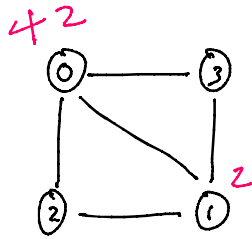
    for (col = 1 → m)
    {
        if (possible → ✓)
        {
            color[node] = col;
            if (f(node + 1) == T)
                return T

            color[node] = 0; // back track
        }
    }

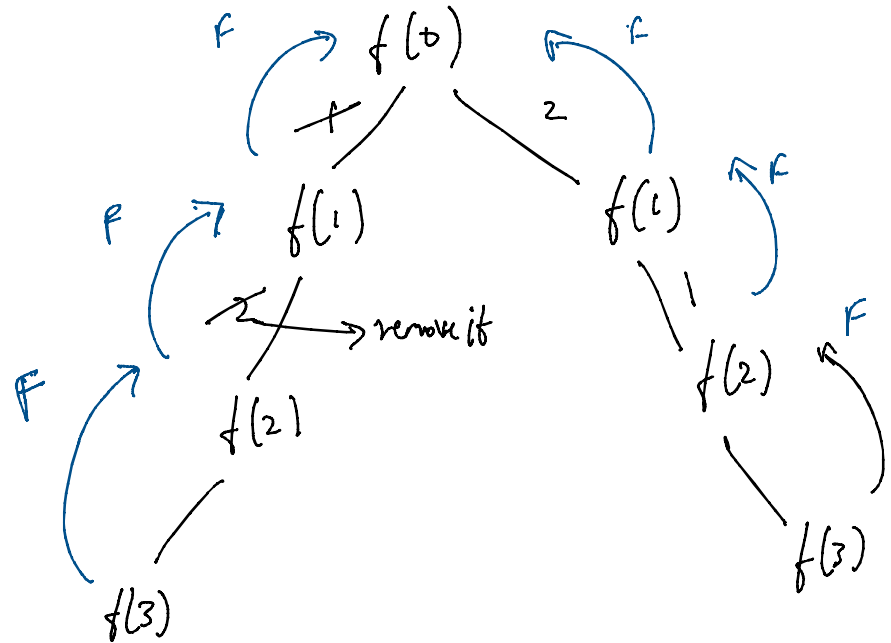
    return false.
}

```

$$M = 2 \quad N = 4$$



False



$$T.C = O(N^M)$$

$$S.C = O(N)$$

```

11 bool isSafe(int node, int color[], bool graph[101][101], int n, int col) {
12     for(int k = 0; k < n; k++) {
13         if(k != node && graph[k][node] == 1 && color[k] == col) {
14             return false;
15         }
16     }
17     return true;
18 }
19 bool solve(int node, int color[], int m, int N, bool graph[101][101]) {
20     if(node == N) {
21         return true;
22     }
23
24     for(int i = 1; i <= m; i++) {
25         if(isSafe(node, color, graph, N, i)) {
26             color[node] = i;
27             if(solve(node+1, color, m, N, graph)) return true;
28             color[node] = 0;
29         }
30     }
31     return false;
32 }
33 }
34
35 //Function to determine if graph can be coloured with at most M colours such
36 //that no two adjacent vertices of graph are coloured with same colour.
37 bool graphColoring(bool graph[101][101], int m, int N)
38 {
39     int color[N] = {0};
40     if(solve(0, color, m, N, graph)) return true;
41     return false;
42 }

```