Graph coloring problem

let G be a graph and m be a positive integer. We need to find whether the nodes of G can be colored in such a way that no two adjacent nodes have the same color yet only m colors are used

Imp note

if d is the degree of given graph G then, it can be colored with (d+1) colors

Time complexity - O(nm^n)

Approach-

① f^n ①, mColoring(k)
node ↑ k

in an ∞ loop

1) call nextvalue(k) to get valid color
2) return if no color found
3) if node is last, write array x
else go for next node by mcoloring (k+1)

if (k+1) node does not have valid color we check nextvalue(k) for k for valid color for (k+1) node

② f^n ② nextvalue(k)

in an ∞ loop

1) choose next color by using 'mod'
2) if no color chosen, return
3) use loop to travel for each node (other) j in range (1 to n)
no. of nodes ↑ n
    i) check if adjacent } if both true, break, this
    ii) color same } will go to check for next color → step 1
4) if step 3, does not break
checked by ~~k~~ j = (n+1) then return

to check validity of each new color if previous color found to be same as adjacent

m= no. of colors (1,2,3)
x - global array, solution in n-tuple, index → node, value → color number
K - current node (0,1,2,3)
n - no. of nodes.

```
x = [0]*n

mColoring (K)
{
    repeat
      { nextValue(K)                          # next color, stored in x[K]
        if (x[K]==0) then return              # no valid color, x[K]==0
        if (K==n-1)then write x([1:n])        # K is 0 indexed so, if k==n-1
                                                it means k is last node
        else
            mColoring(K+1)                    # move to next node
      } until (false)                         # x[K]==0, terminates this
}                                                 ∞ loop

nextValue (K)
{
    repeat
      {                                       # for each loop, x[K] is tried
          x[K] = (x[K]+1)mod(m+1)             # for valid color
          if (x[K]==0) then return            # no valid color found
          for j ← 0 to n-1                    # for each available node
              do if (G[j][K]≠0 and              # if adjacent
                     x[j] == x[K])              # same color
                  then break

        if (j== n-1) then return              # if for loop executes without
      } until (false)                           break then the color is valid
}                                               hence return.
```