Devang Chhajed

# Colouring Graph problem using Backtracking

**Code:**

```
import turtle
import math


totalnodes = 0
totalcolors = 0
graph=[]
mcolorlist=[]
colorlist=[]
nodecolor=[]

def initgraph():
#       totalnodes=int(input("Total Nodes : "))
#       totalcolors=int(input("Total Colors : "))
        for i in range(0, totalnodes):
        a=set()
        graph.append(a)
        nodecolor.append(-1)

        for i in range(0, totalcolors):
        c = input("Enter colors : ")
        mcolorlist.append(c)


def enteredge():
        te = int(input("Enter total Edge : "))
        for i in range(0, te):
        n1, n2 = map(int,input().split())
        graph[n1].add(n2)
        graph[n2].add(n1)


def check(v, c):
        for i in graph[v]:
        if nodecolor[i]!=-1 and nodecolor[i]==c:
        return False

        return True
```

Devang Chhajed

```
def sol(v):
        if v>totalnodes:
        return False
        if -1 not in nodecolor and len(nodecolor)>0:
        return True

#       print("Starting loop for ",v)
        for colr in range(0, totalcolors):
#       print("check",v,colr)
        if check(v, colr):
        nodecolor[v] = colr
#       print(nodecolor)
        if sol(v+1)==True:
                return True
        else:
                nodecolor[v] = -1
#       print(nodecolor,"--")

        return False


def mainsol():
#       print(totalcolors)
        """
        colorlist=mcolorlist
        if sol(0)==False:
        print("Solution does not Exist")
        else:
        for i in range(0,totalnodes):
        print("Node ",i," : ", colorlist[nodecolor[i]])
        #print(nodecolor)
        """

        i=1
        flag = False
        while flag==False:
        totalcolors = i
        #colorlist.clear()
        print("---",totalnodes)
        nodecolor=[-1]*totalnodes
        print(nodecolor)
        colorlist=mcolorlist[0:i].copy()
```

```
        flag=sol(0)
        i+=1

        print("Min Color : ", i)
        print(totalnodes)
        print(mcolorlist)
        print(nodecolor)
        for i in range(0,totalnodes):
        print("Node ",i," : ", mcolorlist[nodecolor[i]])




totalnodes=int(input("Total Nodes : "))
totalcolors=int(input("Total Colors : "))

initgraph()
enteredge()
#mainsol()

i=1
flag = False
while flag==False:
        totalcolors = i
        #colorlist.clear()
        nodecolor=[-1]*totalnodes
        colorlist=mcolorlist[0:i].copy()
        flag=sol(0)
        i+=1

print("Min Color : ", i-1)
for i in range(0,totalnodes):
        print("Node ",i," : ", mcolorlist[nodecolor[i]])




#Draw Graph
```

Devang Chhajed

```
radius=150

x=[]
y=[]

for i in range(0,360,360//totalnodes):
        x.append(150*math.cos(math.radians(i)))
        y.append(150*math.sin(math.radians(i)))
print(x)
print(y)

turtle.penup()

for i in range(totalnodes):
        turtle.goto(x[i],y[i])
        turtle.right(90)
        turtle.forward(20)
        turtle.left(90)
        turtle.pendown()
        turtle.color("black",mcolorlist[nodecolor[i]])
        turtle.begin_fill();
        turtle.write(str(int(i+1)),font=("Arial",16, "normal"))
        turtle.circle(20)
        turtle.end_fill();
        turtle.penup()

lineIn=[]
lineOut=[]
"""
for i in range(totalnodes):
        for j in range(i,totalnodes):
        if matrix[i][j]==1:
        lineIn.append(i)
        lineOut.append(j)
"""
for i in range(len(graph)):
        xx = list(graph[i])
        for j in range(len(xx)):
        lineIn.append(i)
        lineOut.append(xx[j])

turtle.penup()
```

Devang Chhajed

```
for i in range(len(lineIn)):
        turtle.goto(x[lineIn[i]],y[lineIn[i]])
        turtle.pendown()
        turtle.goto(x[lineOut[i]],y[lineOut[i]])
        turtle.penup()


turtle.goto(100,-200)
turtle.write("NColouringSystem",font=("Arial",16, "normal"))
```

**Output:**

```
== RESTART: C:/Users/mca_dept/Desktop/backtracking node color single sol.py ==
Total Nodes : 4
Total Colors : 3
Enter colors : red
Enter colors : blue
Enter colors : green
Enter total Edge : 5
0 1
0 2
0 3
2 1
2 3
Min Color :  3
Node  0  :  red
Node  1  :  blue
Node  2  :  green
Node  3  :  blue
```

Devang Chhajed