SUMMER INTERNSHIP 2019

TOPIC-: GRAPH COLOURING

PROFESSOR-: DR.SUBRAHMANYAM KALYANSUNDARAM

Submitted By:- AYUSHI SENGAR

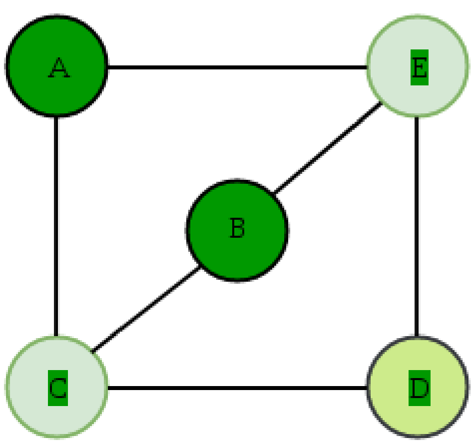
DAYALBAGH EDUCATIONAL INSTITUTE, AGRA

Btech, Computer Science ,2nd Year

GRAPH COLORING: A **graph coloring** is an assignment of labels, called colors, to the vertices of a graph such that no two adjacent vertices share the same color. The **chromatic number** $\chi(G)$ of a graph G is $t$he *minimal* number of colors for which such an assignment is possible.

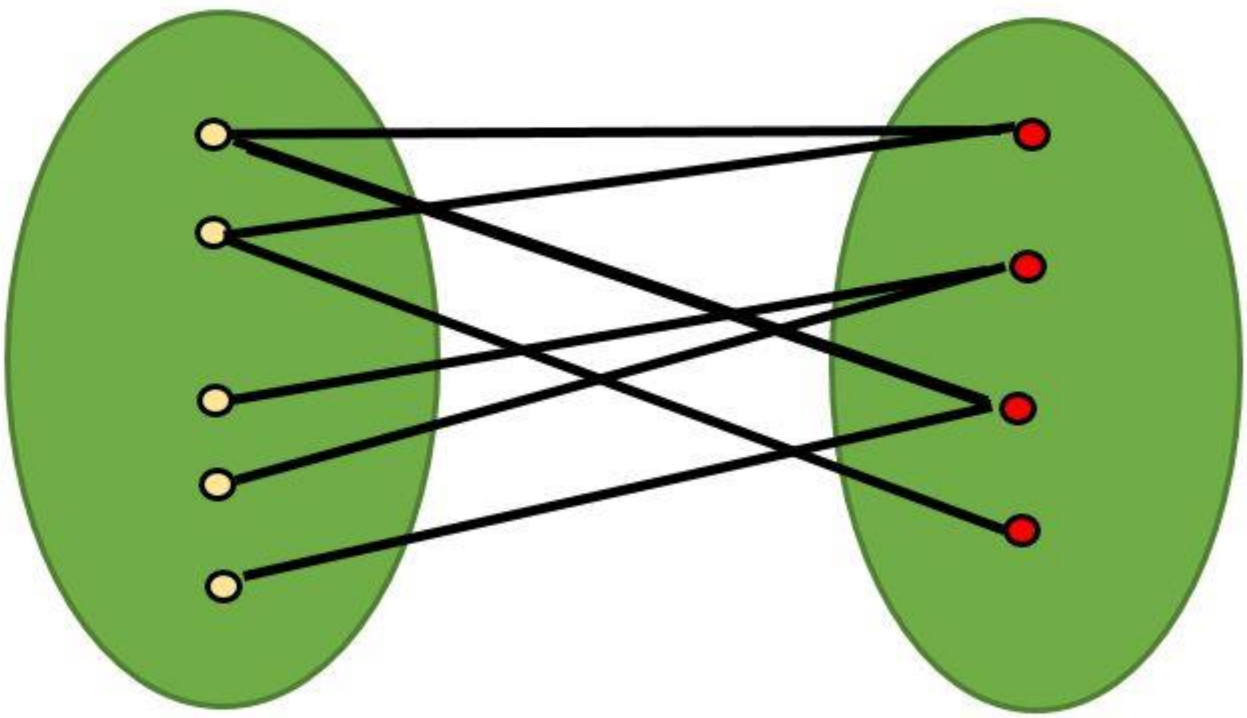
A graph G is called **k-colorable** if there exists a graph coloring on G with k colors. If a graph is k-colorable, then it is n-colorable for any n > k$n>k$. A graph has a chromatic number that is at least as large as the chromatic number of any of its subgraphs. A graph has a chromatic number that is at most one larger than the chromatic number of a subgraph containing only one less vertex.

CHROMATIC NUMBER: The smallest number of colors needed to color a graph G is called its chromatic number. For example, the following can be colored minimum 3 colors.



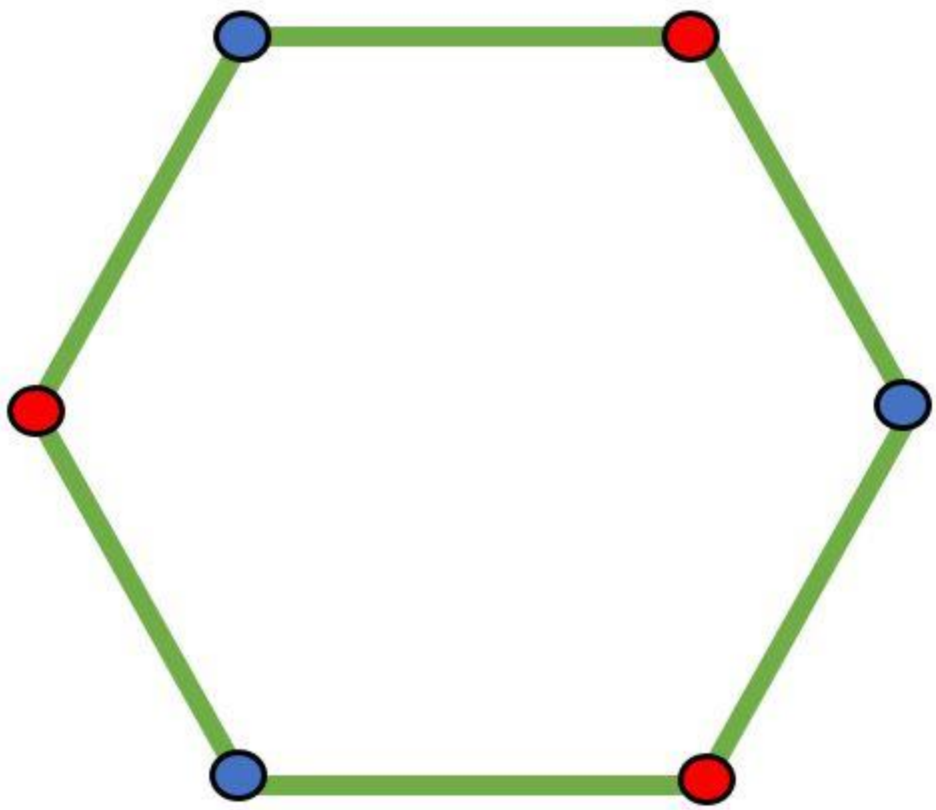
BIPARTITE GRAPHS

Bipartite Graph is a graph whose vertices can be divided into two independent sets, U and V such that every edge (u, v) either connects a vertex from U to V or a vertex from V to U. In other words, for every edge (u, v), either u belongs to U and v to V, or u belongs to V and v to U. We can also say that there is no edge that connects vertices of same set.



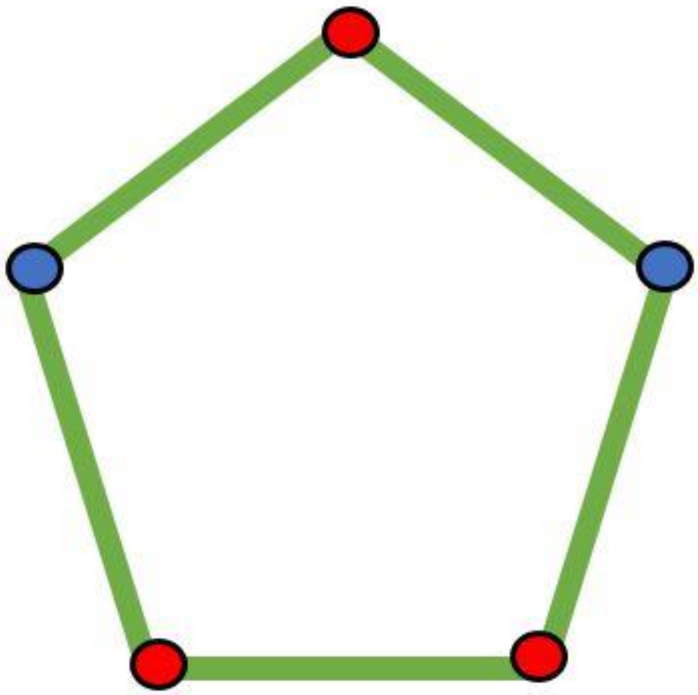
A bipartite graph is possible if the graph coloring is possible using two colors such that vertices in a set are colored with the same color. Note that it is possible to color a cycle graph with even cycle using two colours.

For example, see the following graph.



Cycle graph of length 6

It is not possible to color a cycle graph with odd cycle using two colors.



Cycle graph of length 5

Then we learned the algorithm to check if a graph is bipartite or not by using bfs and dfs techniques.

To check whether a graph is 3 or more colorable is a NP problem i.e. a hard problem so there is no such effecient algorithm for it .

Eventually by gaining the knowledge of coloring, I made an algorithm and implemented it through a python based program to find out the chromatic number for any given graph with n vertices, within a short time. The algorithm ensures that the output chromatic number will give the least possible number of colors required to colour the vertices of the given graph,following the basic laws of vertex colouring strictly.

CONFLICT FREE COLOURING

Each vertex of the graph must be assigned a color so that for each vertex *v* there is at least one color appearing exactly once in the neighborhood of *v*. The goal is to minimize the number of used colors. We consider both the case of closed neighborhoods, when the neighborhood of a node includes the node itself, and the case of open neighborhoods when a node does not belong to its neighborhood.

Similar to n-colorable graphs , algorithm to know the conflict free chromatic number of a graph is also a np problem.Thinking over it ,I made an algorithm to check a graph for conflict free chromatic number 2,and give a proper colour sequence as output, it is yet to be implemented.

K TUPLE COLORING

A k-tuple colouring of a graph G is an assignment of a set of k different colours to each of the vertices of G such that no two adjacent vertices are assigned a common colour. We are supposed to find the smallest positive integer n, such that G has a k-tuple colouring using n colours.

So let's say we have a graph G, such that V = {a, b, c, d, e, f, g} and E = {(a,b), (a,c), (b,d), (c,d), (b,e), (c,f), (e,d), (f,d), (e,g), (f,g), (a,g), (b,c), (e,f)}. The value of k is 2.

Here is a 2-tuple colouring of G with 7 colours
A→A→ red, orange
B→B→ yellow, green
C→C→ blue, indigo
D→D→ red, violet
E→E→ blue, indigo
F→F→ orange, yellow
G→G→ green, violet

<u>APPLICATIONS OF GRAPH COLOURING</u>

The graph coloring problem has huge number of applications:-

1) **<u>Making Schedule or Time Table:</u>**Suppose we want to make am exam schedule for a university. We have list different subjects and students enrolled in every subject. Many subjects would have common students (of same batch, some backlog students, etc). *How do we schedule the exam so that no two exams with a common student are scheduled at same time? How many minimum time slots are needed to schedule all exams?* This problem can be represented as a graph where every vertex is a subject and an edge between two vertices mean there is a common student. So this is a graph coloring problem where minimum number of time slots is equal to the chromatic number of the graph.

2) **<u>Mobile Radio Frequency Assignment:</u>**When frequencies are assigned to towers, frequencies assigned to all towers at the same location must be different. How to assign frequencies with this constraint? What is the minimum number of frequencies needed? This problem is also an instance of graph coloring problem where every tower represents a vertex and an edge between two towers represents that they are in range of each other.

3) **<u>Sudoku:</u>** Sudoku is also a variation of Graph coloring problem where every cell represents a vertex. There is an edge between two vertices if they are in same row or same column or same block.

4) **<u>Register Allocation:</u>** In compiler optimization, register allocation is the process of assigning a large number of target program variables onto a small number of CPU registers. This problem is also a graph coloring problem.

5) **<u>Bipartite Graphs</u>*:*** We can check if a graph is Bipartite or not by coloring the graph using two colors. If a given graph is 2-colorable, then it is Bipartite, otherwise not.

6) **<u>Map Coloring:</u>** Geographical maps of countries or states where no two adjacent cities cannot be assigned same color. Four colors are sufficient to color any map.