Algorithmic Problem Solving

17ECSE309

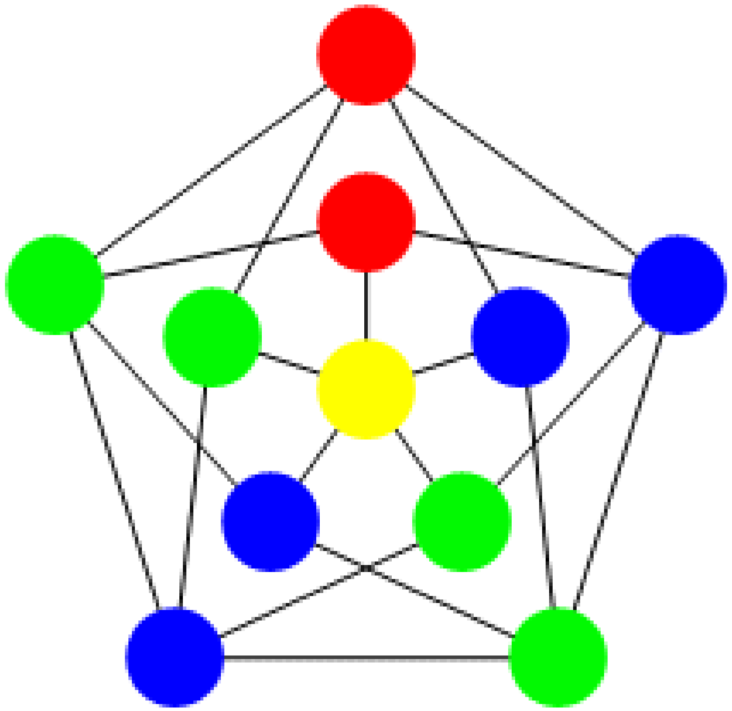
Graph Colouring

-

Veeresh K

What Is Graph Colouring ???

Given an undirected graph $G=(V,E)$, the Graph Colouring Problem (GCP) consists in assigning a colour to each vertex of the graph G in such a way that any two adjacent vertices are assigned different colours, and the number of different colours used is minimized.



Colourability

- **2-colorability**

There is a simple algorithm for determining whether a graph is 2-colorable and assigning colours to its vertices: do a breadth-first search, assigning "red" to the first layer, "blue" to the second layer, "red" to the third layer, etc. Then go over all the edges and check whether the two endpoints of this edge have different colours. This algorithm is $O(|V|+|E|)$ and the last step ensures its correctness.

- **Chromatic Number**

The chromatic number of a graph is the minimum number of colours in a proper colouring of that graph. If chromatic number is K then the graph is K chromatic.

Graph Colouring Algorithm

- There is no efficient algorithm available for colouring a graph with minimum number of colours.
- Graph colouring problem is a known NP Complete problem. (NP complete problems are problems whose status is unknown. No polynomial time algorithm has yet been discovered for any NP complete problem.)
- Although Graph coloring problem is NP Complete problem there are some approximate algorithms to solve the graph coloring problem

Basic Greedy Algorithm

- Colour first vertex with first colour.
- Do following for remaining V-1 vertices.

    Consider the currently picked vertex and colour it with the lowest numbered colour that has not been used on any previously coloured vertices adjacent to it. If all previously used colours appear on vertices adjacent to v, assign a new colour to it.

Welsh Powell Algorithm

- Find the degree of each vertex
- List the vertices in order of descending valence i.e. degree(v(i))>=degree(v(i+1)).
- Colour the first vertex in the list.
- Go down the sorted list and colour every vertex not connected to the coloured vertices above the same colour then cross out all coloured vertices in the list.
- Repeat the process on the uncoloured vertices with a new colour-always working in descending order of degree until all vertices are coloured.
- Complexity of above algorithm = $O(n^2)$

APPLICATIONS

- Sudoku
- Scheduling
- Mobile radio frequency assignment
- Pattern matching
- Register Allocation

References

- https://en.wikipedia.org/wiki/Graph_coloring
- https://www.slideshare.net/PriyankJain26/graph-coloring-48222920
- https://www.slideshare.net/rashikaahuja/graph-coloring-28933847
- https://www.slideshare.net/mohammadalkhalil/graph-coloring-andapplications

THANK YOU