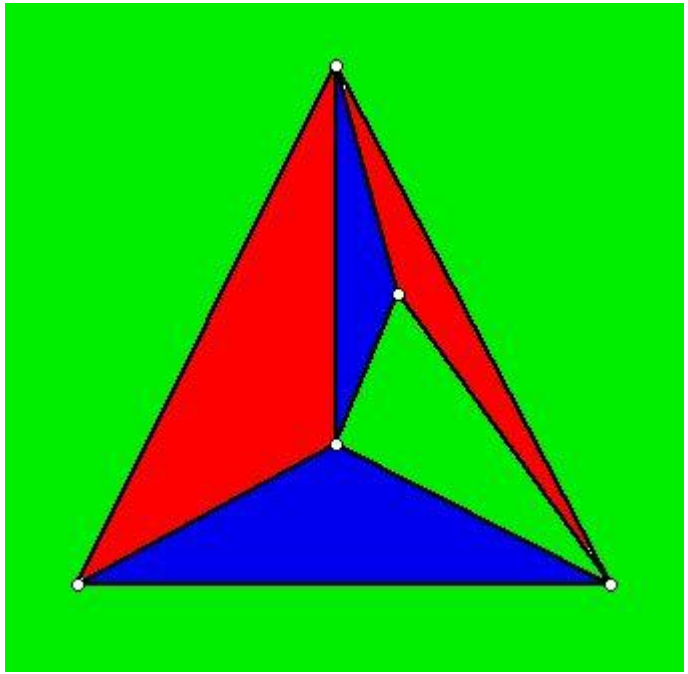
To find out the minimum face-coloring of a given graph

Group 7:
Yash Sharma - IHM2014006
Aditya Kumawat - ICM2014001
Shubham Vashishta - IHM2014007

Problem



Given a graph's list of faces,

- We have to colour all the faces such that no two adjacent faces are of same color.

The solution can be divided into two parts:

1. Converting a given graph to its dual graph.
2. Solving vertex coloring problem for the dual graph

Part 1: Obtaining Dual Graph

Pseudocode:

```
vvi a(n);
vvi b(n, vi(n, 0));
map<ii, set<int> > edgeFaces;

for i=0 to n-1
        int m = a[i].size();
        for j=0 to m-1
                edgeFaces[ii(a[i][j], a[i][(j+1)%m])].push_back(i);

for it=edgeFaces.begin() to edgeFaces.end()
        set<int> ver = it->S;
        if(ver.size()==2)
                b[*ver.begin()][*next(ver.begin())] = 1;
                b[*next(ver.begin())][*ver.begin()] = 1;
```

Time Complexity Analysis:
O(E)

Part 2: Vertex Coloring Greedy Approach

Algorithm

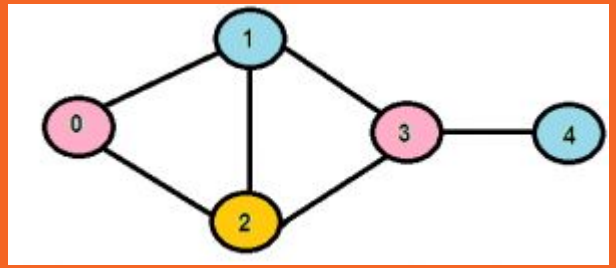
1. Color first vertex with first color.
2. Do following for remaining V-1 vertices.
    a. Consider the currently picked vertex and color it with the lowest numbered color that has not been used on any previously colored vertices adjacent to it. If all previously used colors appear on vertices adjacent to v, assign a new color to it.

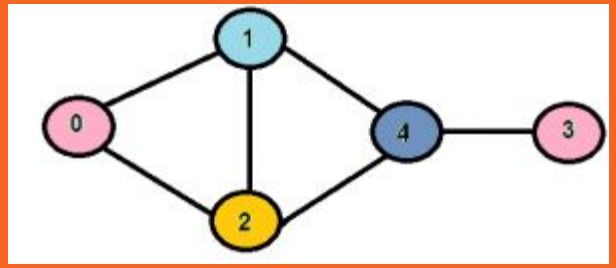
- Doesn't guarantee to use minimum colors
- Guarantees an upper bound on the number of colors.
- Never uses more than (d+1) colors where d is the maximum degree of a vertex in the given graph.

Drawback of Greedy Approach

- Greedy algorithm doesn't always use minimum number of colors.
- The number of colors used sometime depend on the order in which vertices are processed.

  For example, consider the following two graphs. Note that in graph on bottom, vertices 3 and 4 are swapped. If we consider the vertices 0, 1, 2, 3, 4 in top graph, we can color the graph using 3 colors. But if we consider the vertices 0, 1, 2, 3, 4 in bottom graph, we need 4 colors.





Part 2: Vertex Coloring Welsh-Powell Heuristic

1. Find the degree of each vertex.
2. All vertices are sorted according to the decreasing value of their degree in a list V.
3. Colors are ordered in a list C.
4. The first non colored vertex v in V is colored with the first available color in C. Available means a color that was not previously used by the algorithm.
5. The remaining part of the ordered list V is traversed and the same color is allocated to every vertex for which no adjacent vertex has the same color.
6. Steps 4 and 5 are applied iteratively until all the vertices have been colored.

Cont...

1. May not always give the best solution but improvement over greedy approach as it tries to avoid conflicts by picking up first vertices having high degree.

2. Use at most d(G)+1 colors where d(G) represents the largest value of the degree in the graph G.

Conclusion

- The problem of vertex-coloring is NP complete.
- Welsh-Powell heuristic is an improvement over greedy approach as it tries to avoid conflicts by picking up first vertices having high valency.