

31/01/2022.

Expt 2: - Developing agent programs for Real World Programs.

Graph colouring Problem

(*) Theory: - Given an undirected graph and a number m , determine if the graph can be coloured with at most m colours such that no two adjacent vertices have the same colour.

(*) Chromatic Number: - The no. of colours used to colour all the vertices of a graph.

(*) Logic: - Greedy Algorithm is used to solve this problem (Algorithm)

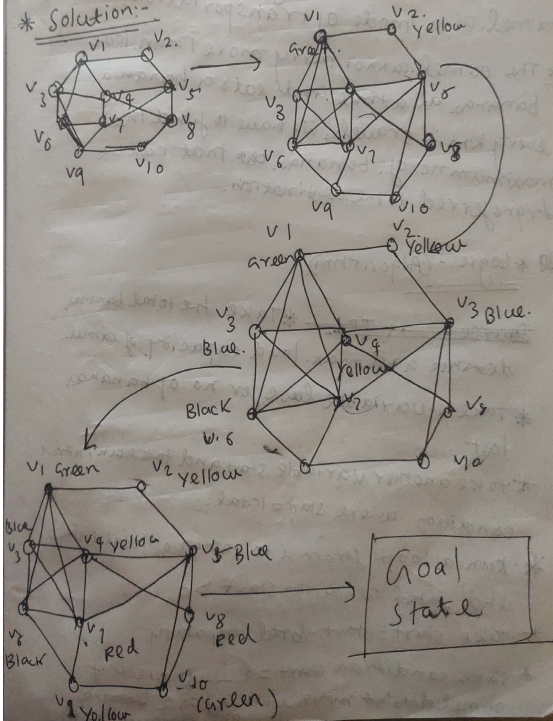
- (1.) Create a recursive function that takes current index, number of vertices, and output colour array
- (2.) If the current index is equal to no. of vertices, check if the output colour configuration is safe, i.e. check if the adjacent vertices do not have same color. If the conditions are met, print the configuration and break.
- (3.) Assign a colour to a vertex. (1 to m)
- (4.) For every assigned colour recursively call the function with next index and no. of vertices
- (5.) If any recursive function returns true, break the loop and return false.

* Problem Formulation:-

* Operators:- The vertices and colours are the operators where the vertices are coloured with the colours provided

* States:- The state is a graph of vertices (uncoloured) to be coloured.

* Solution:-



```
1 class Graph:
2
3     def __init__(self, edges, N):
4         self.adj = [[] for _ in range(N)]
5         for (src, dest) in edges:
6             self.adj[src].append(dest)
7             self.adj[dest].append(src)
8
9     def colorGraph(self):
10         result = {}
11         for u in range(N):
12             assigned = set([result.get(i) for i in graph.adj[u] if i in result])
13             color = 1
14             for c in assigned:
15                 if color != c:
16                     break
17             color = color + 1
18             result[u] = color
19         for v in range(N):
20             print("color assigned to vertex", v, "is", colors[result[v]])
21
22 if __name__ == '__main__':
23     colors = ["", "BLUE", "GREEN", "RED", "YELLOW", "ORANGE", "PINK",
24              "BLACK", "BROWN", "WHITE", "PURPLE", "VIOLET"]
25     edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
26     N = 6
27     graph = Graph(edges, N)
28     colorGraph(graph)
```

```
Color assigned to vertex 0 is BLUE
Color assigned to vertex 1 is GREEN
Color assigned to vertex 2 is BLUE
Color assigned to vertex 3 is RED
Color assigned to vertex 4 is RED
Color assigned to vertex 5 is GREEN

Process exited with code: 0
```

Result:-

The Graph Colouring Problem was successfully implemented using Python