Project 2

Haoran Lyu

For the graph-coloring problem, we want the given graph satisfied that no adjacent nodes have the same color. The constraints in this problem are “No adjacent nodes have the same color” and “We can’t use more than the given color”. Thus, we use the CSP algorithm to solve this problem.

At the beginning, all of the nodes have the domain including every color. Each time we assign a color to a node, the neighbor nodes are constraint to use that color again, so we remove that color from its domain. To determine which node and which color to assign, we use heuristics called min remaining values(MRV) and least constraining value(LCV).

```
125     while len(pass_id) < len(nds):
126         for i in range(len(nds)):
127             if i in pass_id:
128                 continue
129             current = len(nd.nodes[i].domain)
130             if min_num > current:
131                 min_num = current
132                 pick = i
```

This code is for choosing the MRV from all nodes. After we remove the value, we are willing to choose the node with the least domain. Using MRV can reduce the branching factor and it is more likely to cause failure. If we can detect failure in early steps, we can jumpback early in order to reduce the work.

```
32      def color_node(self, id):
33          self.nodes[id].color = min(self.nodes[id].domain)
```

This code is for choosing the LCV from node domain. When using LCV, we expect to use less color to avoid unknown failure. By assigning values that leave maximal flexibility for the remaining variables, we can reach one of the possible success plans earlier.

To detect potential failure earlier, we use AC-3 for constraint propagation. That is, every time we assign a color to a node, we check all the remaining nodes to see if there is a possible solution for assignment. If a node doesn't have any value in its domain because of the constraints, that means the assignment of this color is wrong, and we backtrack to assign another color or choose another node to expand.

```
node: 1 color: 0
node: 2 color: 1
node: 3 color: 2
node: 4 color: 2
node: 5 color: 1
node: 6 color: 0
node: 7 color: 3
```

Here is one solution for the test file.

https://github.com/AranGit2022/graph-coloring-AI