Project Report

On

Constraint satisfaction problems (CSP) - Map Coloring

Project Guidance By

Dr. Dewan Ahmed

Team Details

Sriganesh Lokesh

slokesh1@uncc.edu

801135650

Karthik Rangaraj

krangar2@uncc.edu

801135834

PROBLEM STATEMENT

Constraint Satisfaction Problems consists of a set of objects which satisfies given constraints. Entities are represented as a collection of finite constraints over variables which are then solved. Variants of search techniques such as backtracking, constraint propagation, and local search are used to solve CSP on finite domains.

The CSP functionality functions in a way similar to that of a search tree. The nodes of the search tree contain partial solutions. The goal is to search the tree. A variable is assigned a value which is represented by the branch which is then backtracked later. Branches are pruned in order to avoid dead ends.

CSP is used to solve the map coloring problem. This project demonstrates 4 variants of CSP which we make use of such as:

- **Depth first search only**: This strategy does not prune any branches. All the values assigned so far are checked to see whether they are consistent with each other or not. Variables which are assigned values are examined to ensure the consistency of the values with respect to the constraints. Backtracking is performed if they are not consistent.

- **Depth first search + forward checking**: Wrong options are eliminated from the neighboring variables. In DFS strategy all the variables are examined once a value is assigned to a variable in order to maintain consistency with the constraints. Backtracking is performed if they are not consistent. In FC strategy All its neighbors are checked once a value is assigned to the variable. Incompatible options are eliminated from the neighbors.

- **Depth first search + forward checking + propagation through singleton domains:** Infeasible/impossible options are eliminated from the neighboring states. This variant also makes use of DFS and FC strategy as mentioned previously in Depth first search + forward checking variant, in addition to which makes use of propagation strategy where neighbors are added to the list of variables to propagate depending on whether any options are eliminated from a neighbor in the previous step and if that neighbor has only one variable left. All the variables in the list are propagated.

  Values are crossed off in the neighboring variables which are incompatible with the one remaining option when considering neighbors and its option during propagation of singleton variable. Neighbor is added to list of variables to propagate when the neighbor has one option left after crossing off options in that neighbor.

- **Depth First Search with Heuristic**: This variation performs similar to Depth First Search method. This function makes use of the heuristic in order to decrease the amount of time taken to get our result.

- **Depth First Search with Heuristic and Forward Checking:** This variation performs similar to Depth First Search but implements both heuristic and forward checking. This function makes use of FC with heuristic in order to get our result. Aims to provide a time efficient solution compared to previous methods.

- **Depth First Search with heuristic, Forward Checking and singleton:** This variation aims to provide best solution in least possible time. Any problems in the future would be backtracked. Only one instance of the object is replicated which intern produces other instances. The best result is obtained from the Heuristic function.

PROGRAM STRUCTURE

Global Variables

•**paint_true–**
This is a dictionary used to store the color assignment of each state.
•**backtrack –**
Variable which is used to keep track of the number of backtracks.
•**if_singleton–**
This is used to determine whether singleton is used or not.
•**heuristic–**
This is used to determine whether heuristic is implemented or not.

Functions

- **check –** checking if children are connected to themselves.
- **value_next –** Used to take the next value.
- **color_assign –** Assigning color to the map.
- **dfs –** Used to implement depth first search.
- **decrease –** removing the color for that particular state.
- **decrease_for_forward_check –** decreasing domain for forward checking
- **d_validate –** validating the domain values
- **dfs_forward –** depth first search with forward checking.

OUTPUT

USA

1. DFS

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
1
Color Assigned R -> Texas
Color Assigned Y -> Oklahoma
Color Assigned G -> Hawaii
Color Assigned Y -> SouthDakota
Color Assigned Y -> Utah
Color Assigned G -> Wyoming
Color Assigned B -> Nebraska
Color Assigned G -> RhodeIsland
Color Assigned R -> Maine
Color Assigned G -> NewHampshire
Color Assigned R -> Vermont
Color Assigned B -> Massachusetts
Color Assigned R -> Connecticut
Color Assigned Y -> Tennessee
Color Assigned B -> Virginia
Color Assigned Y -> WestVirginia
Color Assigned G -> Maryland
Color Assigned Y -> Delaware
Color Assigned B -> NewJersey
Color Assigned G -> NewYork
Color Assigned R -> Pennsylvania
Color Assigned B -> Ohio
Color Assigned G -> Kentucky
Color Assigned R -> Indiana
Color Assigned G -> Michigan
Color Assigned R -> Wisconsin
```

```
Color Assigned B -> Illinois
Color Assigned G -> Iowa
Color Assigned B -> Minnesota
Color Assigned R -> NorthDakota
Color Assigned B -> Montana
Color Assigned R -> Idaho
Color Assigned R -> Alaska
Color Assigned G -> Washington
Color Assigned B -> Oregon
Color Assigned G -> Nevada
Color Assigned R -> California
Color Assigned B -> Arizona
Color Assigned G -> NewMexico
Color Assigned R -> Colorado
Color Assigned G -> Kansas
Color Assigned R -> Missouri
Color Assigned B -> Arkansas
Color Assigned G -> Louisiana
Color Assigned R -> Mississippi
Color Assigned G -> Alabama
Color Assigned R -> Florida
Color Assigned B -> Georgia
Color Assigned G -> SouthCarolina
Color Assigned R -> NorthCarolina


Time:  0.37268043200037937
Number of Backtracks 73
```

2. DFS + FC

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
2
Color Assignment in Progress R -> NorthCarolina
```

```
Color Assignment in Progress G -> SouthCarolina
Color Assignment in Progress B -> Georgia
Color Assignment in Progress R -> Florida
Color Assignment in Progress G -> Alabama
Color Assignment in Progress R -> Mississippi
Color Assignment in Progress G -> Louisiana
Color Assignment in Progress B -> Arkansas
Color Assignment in Progress R -> Missouri
Color Assignment in Progress G -> Kansas
Color Assignment in Progress R -> Colorado
Color Assignment in Progress G -> NewMexico
Color Assignment in Progress Y -> Oklahoma
Color Assignment in Progress R -> Texas
Color Assigned R -> Texas
Color Assigned Y -> Oklahoma
Color Assignment in Progress B -> Arizona
Color Assignment in Progress R -> California
Color Assignment in Progress G -> Hawaii
Color Assigned G -> Hawaii
Color Assignment in Progress G -> Nevada
Color Assignment in Progress B -> Oregon
Color Assignment in Progress R -> Washington
Color Assignment in Progress G -> Washington
Color Assignment in Progress R -> Idaho
Color Assignment in Progress G -> Montana
Color Assignment in Progress R -> NorthDakota
Color Assignment in Progress G -> Minnesota
Color Assignment in Progress B -> Iowa
Color Assignment in Progress Y -> Iowa
Color Assignment in Progress B -> Minnesota
Color Assignment in Progress G -> Iowa
Color Assignment in Progress B -> Nebraska
Color Assignment in Progress Y -> Minnesota
Color Assignment in Progress G -> Iowa
Color Assignment in Progress Y -> Nebraska
Color Assignment in Progress B -> NorthDakota
Color Assignment in Progress R -> Minnesota
Color Assignment in Progress G -> Iowa
Color Assignment in Progress B -> Nebraska
Color Assignment in Progress B -> Iowa
Color Assignment in Progress G -> Minnesota
Color Assignment in Progress B -> Iowa
Color Assignment in Progress Y -> Nebraska
Color Assignment in Progress B -> Wyoming
Color Assignment in Progress R -> SouthDakota
Color Assigned R -> SouthDakota
Color Assignment in Progress Y -> Utah
Color Assigned Y -> Utah
```

```
Color Assigned B -> Wyoming
Color Assigned Y -> Nebraska
Color Assignment in Progress G -> Illinois
Color Assignment in Progress R -> Wisconsin
Color Assignment in Progress B -> Michigan
Color Assignment in Progress R -> Indiana
Color Assignment in Progress B -> Kentucky
Color Assignment in Progress G -> Ohio
Color Assignment in Progress R -> Pennsylvania
Color Assignment in Progress G -> NewYork
Color Assignment in Progress R -> Connecticut
Color Assignment in Progress B -> Massachusetts
Color Assignment in Progress G -> RhodeIsland
Color Assigned G -> RhodeIsland
Color Assignment in Progress R -> Vermont
Color Assignment in Progress G -> NewHampshire
Color Assignment in Progress R -> Maine
Color Assigned R -> Maine
Color Assigned G -> NewHampshire
Color Assigned R -> Vermont
Color Assigned B -> Massachusetts
Color Assigned R -> Connecticut
Color Assignment in Progress B -> NewJersey
Color Assignment in Progress G -> Delaware
Color Assignment in Progress B -> Maryland
Color Assignment in Progress Y -> WestVirginia
Color Assignment in Progress G -> Virginia
Color Assignment in Progress Y -> Tennessee
Color Assigned Y -> Tennessee
Color Assigned G -> Virginia
Color Assigned Y -> WestVirginia
Color Assigned B -> Maryland
Color Assigned G -> Delaware
Color Assigned B -> NewJersey
Color Assigned G -> NewYork
Color Assigned R -> Pennsylvania
Color Assigned G -> Ohio
Color Assigned B -> Kentucky
Color Assigned R -> Indiana
Color Assigned B -> Michigan
Color Assigned R -> Wisconsin
Color Assigned G -> Illinois
Color Assigned B -> Iowa
Color Assigned G -> Minnesota
Color Assigned B -> NorthDakota
Color Assigned G -> Montana
Color Assigned R -> Idaho
Color Assignment in Progress R -> Alaska
```

```
Color Assigned R -> Alaska
Color Assigned G -> Washington
Color Assigned B -> Oregon
Color Assigned G -> Nevada
Color Assigned R -> California
Color Assigned B -> Arizona
Color Assigned G -> NewMexico
Color Assigned R -> Colorado
Color Assigned G -> Kansas
Color Assigned R -> Missouri
Color Assigned B -> Arkansas
Color Assigned G -> Louisiana
Color Assigned R -> Mississippi
Color Assigned G -> Alabama
Color Assigned R -> Florida
Color Assigned B -> Georgia
Color Assigned G -> SouthCarolina
Color Assigned R -> NorthCarolina

Time:  0.4289514929996585
Number of Backtracks 15
```

3. DFS + FC + Singleton

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
3
Color Assignment in Progress R -> NorthCarolina
Color Assignment in Progress G -> SouthCarolina
Color Assignment in Progress B -> Georgia
Color Assignment in Progress G -> Tennessee
Color Assignment in Progress B -> Virginia
Color Assignment in Progress R -> Kentucky
Color Assignment in Progress B -> Missouri
Color Assignment in Progress R -> Arkansas
```

```
Color Assignment in Progress G -> Oklahoma
Color Assignment in Progress R -> Kansas
Color Assignment in Progress B -> Colorado
Color Assignment in Progress G -> Nebraska
Color Assignment in Progress R -> Iowa
Color Assignment in Progress G -> Illinois
Color Assignment in Progress B -> Wisconsin
Color Assignment in Progress G -> Minnesota
Color Assignment in Progress B -> SouthDakota
Color Assignment in Progress R -> Wyoming
Color Assignment in Progress G -> Montana
Color Assignment in Progress R -> NorthDakota
Color Assigned R -> NorthDakota
Color Assignment in Progress B -> Idaho
Color Assignment in Progress G -> Utah
Color Assignment in Progress R -> NewMexico
Color Assignment in Progress Y -> Arizona
Color Assignment in Progress R -> Nevada
Color Assignment in Progress G -> California
Color Assignment in Progress Y -> Oregon
Color Assignment in Progress R -> Washington
Color Assignment in Progress G -> Alaska
Color Assigned G -> Alaska
Color Assigned R -> Washington
Color Assigned Y -> Oregon
Color Assignment in Progress R -> Hawaii
Color Assigned R -> Hawaii
Color Assigned G -> California
Color Assigned R -> Nevada
Color Assigned Y -> Arizona
Color Assignment in Progress B -> Texas
Color Assignment in Progress G -> Louisiana
Color Assignment in Progress B -> Mississippi
Color Assignment in Progress R -> Alabama
Color Assignment in Progress G -> Florida
Color Assigned G -> Florida
Color Assigned R -> Alabama
Color Assigned B -> Mississippi
Color Assigned G -> Louisiana
Color Assigned B -> Texas
Color Assigned R -> NewMexico
Color Assigned G -> Utah
Color Assigned B -> Idaho
Color Assigned G -> Montana
Color Assigned R -> Wyoming
Color Assigned B -> SouthDakota
Color Assigned G -> Minnesota
Color Assignment in Progress R -> Michigan
```

```
Color Assignment in Progress B -> Indiana
Color Assignment in Progress G -> Ohio
Color Assignment in Progress Y -> WestVirginia
Color Assignment in Progress R -> Maryland
Color Assignment in Progress B -> Pennsylvania
Color Assignment in Progress G -> Delaware
Color Assignment in Progress R -> NewJersey
Color Assignment in Progress G -> NewYork
Color Assignment in Progress R -> Connecticut
Color Assignment in Progress B -> Massachusetts
Color Assignment in Progress G -> RhodeIsland
Color Assigned G -> RhodeIsland
Color Assignment in Progress R -> Vermont
Color Assignment in Progress G -> NewHampshire
Color Assignment in Progress R -> Maine
Color Assigned R -> Maine
Color Assigned G -> NewHampshire
Color Assigned R -> Vermont
Color Assigned B -> Massachusetts
Color Assigned R -> Connecticut
Color Assigned G -> NewYork
Color Assigned R -> NewJersey
Color Assigned G -> Delaware
Color Assigned B -> Pennsylvania
Color Assigned R -> Maryland
Color Assigned Y -> WestVirginia
Color Assigned G -> Ohio
Color Assigned B -> Indiana
Color Assigned R -> Michigan
Color Assigned B -> Wisconsin
Color Assigned G -> Illinois
Color Assigned R -> Iowa
Color Assigned G -> Nebraska
Color Assigned B -> Colorado
Color Assigned R -> Kansas
Color Assigned G -> Oklahoma
Color Assigned R -> Arkansas
Color Assigned B -> Missouri
Color Assigned R -> Kentucky
Color Assigned B -> Virginia
Color Assigned G -> Tennessee
Color Assigned B -> Georgia
Color Assigned G -> SouthCarolina
Color Assigned R -> NorthCarolina


Time:  0.41947213700041175
Number of Backtracks 0
```

4. DFS + Heuristic

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
4
[(-1, -1, 'SouthCarolina'), (-1, -4, 'Virginia'), (-1, -7, 'Tennessee'), (-1, -4, 'Georgi
a')] ()()()()()
[(-1, -7, 'Tennessee'), (-1, -4, 'Georgia'), (-1, -4, 'Virginia'), (-1, -1, 'SouthCarolin
a')] --Sort - ()()()()()
[(-1, -5, 'Arkansas'), (-1, -7, 'Missouri'), (-1, -6, 'Kentucky'), (-1, -3, 'Mississippi'
), (-2, -3, 'Virginia'), (-1, -3, 'Alabama'), (-2, -3, 'Georgia')] ()()()()()
[(-2, -3, 'Georgia'), (-2, -3, 'Virginia'), (-1, -7, 'Missouri'), (-1, -6, 'Kentucky'), (
-1, -5, 'Arkansas'), (-1, -3, 'Alabama'), (-1, -3, 'Mississippi')] --Sort - ()()()()()
[(-1, -1, 'Florida'), (-2, 0, 'SouthCarolina'), (-2, -2, 'Alabama')] ()()()()()
[(-2, -2, 'Alabama'), (-2, 0, 'SouthCarolina'), (-1, -1, 'Florida')] --Sort - ()()()()()
[(-2, 0, 'Florida'), (-2, -2, 'Mississippi')] ()()()()()
[(-2, -2, 'Mississippi'), (-2, 0, 'Florida')] --Sort - ()()()()()
[(-1, -2, 'Louisiana'), (-2, -4, 'Arkansas')] ()()()()()
[(-2, -4, 'Arkansas'), (-1, -2, 'Louisiana')] --Sort - ()()()()()
[(-2, -6, 'Missouri'), (-1, -5, 'Oklahoma'), (-1, -3, 'Texas'), (-2, -1, 'Louisiana')] ()
()()()()
[(-2, -6, 'Missouri'), (-2, -1, 'Louisiana'), (-1, -5, 'Oklahoma'), (-1, -3, 'Texas')] --
Sort - ()()()()()
[(-1, -3, 'Kansas'), (-1, -5, 'Iowa'), (-2, -4, 'Oklahoma'), (-1, -5, 'Nebraska'), (-1, -
5, 'Illinois'), (-2, -5, 'Kentucky')] ()()()()()
[(-2, -5, 'Kentucky'), (-2, -4, 'Oklahoma'), (-1, -5, 'Illinois'), (-1, -5, 'Iowa'), (-1,
-5, 'Nebraska'), (-1, -3, 'Kansas')] --Sort - ()()()()()
[(-2, -4, 'Illinois'), (-1, -4, 'Ohio'), (-1, -3, 'Indiana'), (-2, -2, 'Virginia'), (-1,
-4, 'WestVirginia')] ()()()()()
[(-2, -4, 'Illinois'), (-2, -2, 'Virginia'), (-1, -4, 'Ohio'), (-1, -4, 'WestVirginia'),
(-1, -3, 'Indiana')] --Sort - ()()()()()
[(-2, -4, 'Iowa'), (-1, -3, 'Wisconsin'), (-1, -3, 'Michigan'), (-2, -2, 'Indiana')] ()()
()()()
```

```
[(-2, -4, 'Iowa'), (-2, -2, 'Indiana'), (-1, -3, 'Michigan'), (-1, -3, 'Wisconsin')] --So
rt - ()()()()()
[(-2, -4, 'Nebraska'), (-1, -5, 'SouthDakota'), (-2, -2, 'Wisconsin'), (-1, -3, 'Minnesot
a')] ()()()()()
[(-2, -4, 'Nebraska'), (-2, -2, 'Wisconsin'), (-1, -5, 'SouthDakota'), (-1, -3, 'Minnesot
a')] --Sort - ()()()()()
[(-1, -6, 'Colorado'), (-2, -2, 'Kansas'), (-1, -5, 'Wyoming'), (-2, -4, 'SouthDakota')]
()()()()()
[(-2, -4, 'SouthDakota'), (-2, -2, 'Kansas'), (-1, -6, 'Colorado'), (-1, -5, 'Wyoming')]
--Sort - ()()()()()
[(-1, -3, 'Montana'), (-2, -4, 'Wyoming'), (-1, -2, 'NorthDakota'), (-2, -2, 'Minnesota')
] ()()()()()
[(-2, -4, 'Wyoming'), (-2, -2, 'Minnesota'), (-1, -3, 'Montana'), (-1, -2, 'NorthDakota')
] --Sort - ()()()()()
[(-2, -5, 'Colorado'), (-2, -2, 'Montana'), (-1, -5, 'Idaho'), (-1, -5, 'Utah')] ()()()()
()
[(-2, -5, 'Colorado'), (-2, -2, 'Montana'), (-1, -5, 'Idaho'), (-1, -5, 'Utah')] --Sort -
()()()()()
[(-1, -4, 'NewMexico'), (-2, -1, 'Kansas'), (-2, -3, 'Oklahoma'), (-1, -4, 'Arizona'), (-
2, -4, 'Utah')] ()()()()()
[(-2, -4, 'Utah'), (-2, -3, 'Oklahoma'), (-2, -1, 'Kansas'), (-1, -4, 'Arizona'), (-1, -4
, 'NewMexico')] --Sort - ()()()()()
[(-2, -3, 'NewMexico'), (-1, -4, 'Nevada'), (-2, -3, 'Arizona'), (-2, -4, 'Idaho')] ()()(
)()()
[(-2, -4, 'Idaho'), (-2, -3, 'Arizona'), (-2, -3, 'NewMexico'), (-1, -4, 'Nevada')] --Sor
t - ()()()()()
[(-2, -1, 'Montana'), (-2, -3, 'Nevada'), (-1, -2, 'Washington'), (-1, -3, 'Oregon')] ()(
)()()()
[(-2, -3, 'Nevada'), (-2, -1, 'Montana'), (-1, -3, 'Oregon'), (-1, -2, 'Washington')] --S
ort - ()()()()()
[(-1, -3, 'California'), (-3, -2, 'Arizona'), (-2, -2, 'Oregon')] ()()()()()
[(-3, -2, 'Arizona'), (-2, -2, 'Oregon'), (-1, -3, 'California')] --Sort - ()()()()()
[(-3, -2, 'NewMexico'), (-2, -2, 'California')] ()()()()()
[(-3, -2, 'NewMexico'), (-2, -2, 'California')] --Sort - ()()()()()
[(-2, -2, 'Oklahoma'), (-1, -2, 'Texas')] ()()()()()
[(-2, -2, 'Oklahoma'), (-1, -2, 'Texas')] --Sort - ()()()()()
[(-2, 0, 'Kansas'), (-2, -1, 'Texas')] ()()()()()
[(-2, -1, 'Texas'), (-2, 0, 'Kansas')] --Sort - ()()()()()
[(-2, 0, 'Louisiana')] ()()()()()
[(-2, 0, 'Louisiana')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Louisiana
Color Assigned B -> Texas
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Kansas
Color Assigned G -> Oklahoma
```

```
Color Assigned R -> NewMexico
[(-1, 0, 'Hawaii'), (-3, -1, 'Oregon')] ()()()()()
[(-3, -1, 'Oregon'), (-1, 0, 'Hawaii')] --Sort - ()()()()()
[(-2, -1, 'Washington')] ()()()()()
[(-2, -1, 'Washington')] --Sort - ()()()()()
[(-1, 0, 'Alaska')] ()()()()()
[(-1, 0, 'Alaska')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Alaska
Color Assigned R -> Washington
Color Assigned Y -> Oregon
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Hawaii
Color Assigned G -> California
Color Assigned Y -> Arizona
Color Assigned R -> Nevada
[(-2, -1, 'NorthDakota')] ()()()()()
[(-2, -1, 'NorthDakota')] --Sort - ()()()()()
[(-2, -1, 'Minnesota')] ()()()()()
[(-2, -1, 'Minnesota')] --Sort - ()()()()()
[(-2, -1, 'Wisconsin')] ()()()()()
[(-2, -1, 'Wisconsin')] --Sort - ()()()()()
[(-2, -2, 'Michigan')] ()()()()()
[(-2, -2, 'Michigan')] --Sort - ()()()()()
[(-2, -1, 'Indiana'), (-1, -3, 'Ohio')] ()()()()()
[(-2, -1, 'Indiana'), (-1, -3, 'Ohio')] --Sort - ()()()()()
[(-2, -2, 'Ohio')] ()()()()()
[(-2, -2, 'Ohio')] --Sort - ()()()()()
[(-1, -5, 'Pennsylvania'), (-2, -3, 'WestVirginia')] ()()()()()
[(-2, -3, 'WestVirginia'), (-1, -5, 'Pennsylvania')] --Sort - ()()()()()
[(-1, -3, 'Maryland'), (-3, -1, 'Virginia'), (-2, -4, 'Pennsylvania')] ()()()()()
[(-3, -1, 'Virginia'), (-2, -4, 'Pennsylvania'), (-1, -3, 'Maryland')] --Sort - ()()()()(
)
[(-2, -2, 'Maryland')] ()()()()()
[(-2, -2, 'Maryland')] --Sort - ()()()()()
[(-1, -2, 'Delaware'), (-3, -3, 'Pennsylvania')] ()()()()()
[(-3, -3, 'Pennsylvania'), (-1, -2, 'Delaware')] --Sort - ()()()()()
[(-1, -4, 'NewYork'), (-1, -2, 'NewJersey'), (-2, -1, 'Delaware')] ()()()()()
[(-2, -1, 'Delaware'), (-1, -4, 'NewYork'), (-1, -2, 'NewJersey')] --Sort - ()()()()()
[(-2, -1, 'NewJersey')] ()()()()()
[(-2, -1, 'NewJersey')] --Sort - ()()()()()
[(-2, -3, 'NewYork')] ()()()()()
[(-2, -3, 'NewYork')] --Sort - ()()()()()
[(-1, -2, 'Connecticut'), (-1, -2, 'Vermont'), (-1, -4, 'Massachusetts')] ()()()()()
[(-1, -4, 'Massachusetts'), (-1, -2, 'Connecticut'), (-1, -2, 'Vermont')] --Sort - ()()()
()()
```

```
[(-2, -1, 'Connecticut'), (-1, -1, 'RhodeIsland'), (-2, -1, 'Vermont'), (-1, -2, 'NewHamp
shire')] ()()()()()
[(-2, -1, 'Connecticut'), (-2, -1, 'Vermont'), (-1, -2, 'NewHampshire'), (-1, -1, 'RhodeI
sland')] --Sort - ()()()()()
[(-2, 0, 'RhodeIsland')] ()()()()()
[(-2, 0, 'RhodeIsland')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> RhodeIsland
Color Assigned B -> Connecticut
[(-2, -1, 'NewHampshire')] ()()()()()
[(-2, -1, 'NewHampshire')] --Sort - ()()()()()
[(-1, 0, 'Maine')] ()()()()()
[(-1, 0, 'Maine')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Maine
Color Assigned G -> NewHampshire
Color Assigned B -> Vermont
Color Assigned R -> Massachusetts
Color Assigned G -> NewYork
Color Assigned R -> NewJersey
Color Assigned G -> Delaware
Color Assigned Y -> Pennsylvania
Color Assigned R -> Maryland
Color Assigned Y -> Virginia
Color Assigned B -> WestVirginia
Color Assigned G -> Ohio
Color Assigned B -> Indiana
Color Assigned R -> Michigan
Color Assigned B -> Wisconsin
Color Assigned G -> Minnesota
Color Assigned R -> NorthDakota
Color Assigned G -> Montana
Color Assigned B -> Idaho
Color Assigned G -> Utah
Color Assigned B -> Colorado
Color Assigned R -> Wyoming
Color Assigned B -> SouthDakota
Color Assigned G -> Nebraska
Color Assigned R -> Iowa
Color Assigned G -> Illinois
Color Assigned R -> Kentucky
Color Assigned B -> Missouri
Color Assigned R -> Arkansas
Color Assigned B -> Mississippi
[] ()()()()()
[] --Sort - ()()()()()
```

```
Color Assigned G -> Florida
Color Assigned R -> Alabama
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> SouthCarolina
Color Assigned B -> Georgia
Color Assigned G -> Tennessee
Color Assigned R -> NorthCarolina


Time:  0.4287903370004642
Number of Backtracks 0
```

5. DFS + Heuristic + FC

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
5
Color Assignment in Progress R -> NorthCarolina
[(-1, -1, 'SouthCarolina'), (-1, -4, 'Virginia'), (-1, -7, 'Tennessee'), (-1, -4, 'Georgi
a')] ()()()()()
[(-1, -7, 'Tennessee'), (-1, -4, 'Georgia'), (-1, -4, 'Virginia'), (-1, -1, 'SouthCarolin
a')] --Sort - ()()()()()
Color Assignment in Progress G -> Tennessee
[(-1, -5, 'Arkansas'), (-1, -7, 'Missouri'), (-1, -6, 'Kentucky'), (-1, -3, 'Mississippi'
), (-2, -3, 'Virginia'), (-1, -3, 'Alabama'), (-2, -3, 'Georgia')] ()()()()()
[(-2, -3, 'Georgia'), (-2, -3, 'Virginia'), (-1, -7, 'Missouri'), (-1, -6, 'Kentucky'), (
-1, -5, 'Arkansas'), (-1, -3, 'Alabama'), (-1, -3, 'Mississippi')] --Sort - ()()()()()
Color Assignment in Progress B -> Georgia
[(-1, -1, 'Florida'), (-2, 0, 'SouthCarolina'), (-2, -2, 'Alabama')] ()()()()()
[(-2, -2, 'Alabama'), (-2, 0, 'SouthCarolina'), (-1, -1, 'Florida')] --Sort - ()()()()()
Color Assignment in Progress R -> Alabama
[(-2, 0, 'Florida'), (-2, -2, 'Mississippi')] ()()()()()
[(-2, -2, 'Mississippi'), (-2, 0, 'Florida')] --Sort - ()()()()()
Color Assignment in Progress B -> Mississippi
```

```
[(-1, -2, 'Louisiana'), (-2, -4, 'Arkansas')] ()()()()()
[(-2, -4, 'Arkansas'), (-1, -2, 'Louisiana')] --Sort - ()()()()()
Color Assignment in Progress R -> Arkansas
[(-2, -6, 'Missouri'), (-1, -5, 'Oklahoma'), (-1, -3, 'Texas'), (-2, -1, 'Louisiana')] ()
()()()()
[(-2, -6, 'Missouri'), (-2, -1, 'Louisiana'), (-1, -5, 'Oklahoma'), (-1, -3, 'Texas')] --
Sort - ()()()()()
Color Assignment in Progress B -> Missouri
[(-1, -3, 'Kansas'), (-1, -5, 'Iowa'), (-2, -4, 'Oklahoma'), (-1, -5, 'Nebraska'), (-1, -
5, 'Illinois'), (-2, -5, 'Kentucky')] ()()()()()
[(-2, -5, 'Kentucky'), (-2, -4, 'Oklahoma'), (-1, -5, 'Illinois'), (-1, -5, 'Iowa'), (-1,
-5, 'Nebraska'), (-1, -3, 'Kansas')] --Sort - ()()()()()
Color Assignment in Progress R -> Kentucky
[(-2, -4, 'Illinois'), (-1, -4, 'Ohio'), (-1, -3, 'Indiana'), (-2, -2, 'Virginia'), (-1,
-4, 'WestVirginia')] ()()()()()
[(-2, -4, 'Illinois'), (-2, -2, 'Virginia'), (-1, -4, 'Ohio'), (-1, -4, 'WestVirginia'),
(-1, -3, 'Indiana')] --Sort - ()()()()()
Color Assignment in Progress G -> Illinois
[(-2, -4, 'Iowa'), (-1, -3, 'Wisconsin'), (-1, -3, 'Michigan'), (-2, -2, 'Indiana')] ()()
()()()
[(-2, -4, 'Iowa'), (-2, -2, 'Indiana'), (-1, -3, 'Michigan'), (-1, -3, 'Wisconsin')] --So
rt - ()()()()()
Color Assignment in Progress R -> Iowa
[(-2, -4, 'Nebraska'), (-1, -5, 'SouthDakota'), (-2, -2, 'Wisconsin'), (-1, -3, 'Minnesot
a')] ()()()()()
[(-2, -4, 'Nebraska'), (-2, -2, 'Wisconsin'), (-1, -5, 'SouthDakota'), (-1, -3, 'Minnesot
a')] --Sort - ()()()()()
Color Assignment in Progress G -> Nebraska
[(-1, -6, 'Colorado'), (-2, -2, 'Kansas'), (-1, -5, 'Wyoming'), (-2, -4, 'SouthDakota')]
()()()()()
[(-2, -4, 'SouthDakota'), (-2, -2, 'Kansas'), (-1, -6, 'Colorado'), (-1, -5, 'Wyoming')]
--Sort - ()()()()()
Color Assignment in Progress B -> SouthDakota
[(-1, -3, 'Montana'), (-2, -4, 'Wyoming'), (-1, -2, 'NorthDakota'), (-2, -2, 'Minnesota')
] ()()()()()
[(-2, -4, 'Wyoming'), (-2, -2, 'Minnesota'), (-1, -3, 'Montana'), (-1, -2, 'NorthDakota')
] --Sort - ()()()()()
Color Assignment in Progress R -> Wyoming
[(-2, -5, 'Colorado'), (-2, -2, 'Montana'), (-1, -5, 'Idaho'), (-1, -5, 'Utah')] ()()()()
()
[(-2, -5, 'Colorado'), (-2, -2, 'Montana'), (-1, -5, 'Idaho'), (-1, -5, 'Utah')] --Sort -
()()()()()
Color Assignment in Progress B -> Colorado
[(-1, -4, 'NewMexico'), (-2, -1, 'Kansas'), (-2, -3, 'Oklahoma'), (-1, -4, 'Arizona'), (-
2, -4, 'Utah')] ()()()()()
[(-2, -4, 'Utah'), (-2, -3, 'Oklahoma'), (-2, -1, 'Kansas'), (-1, -4, 'Arizona'), (-1, -4
, 'NewMexico')] --Sort - ()()()()()
Color Assignment in Progress G -> Utah
```

```
[(-2, -3, 'NewMexico'), (-1, -4, 'Nevada'), (-2, -3, 'Arizona'), (-2, -4, 'Idaho')] ()()()()()
[(-2, -4, 'Idaho'), (-2, -3, 'Arizona'), (-2, -3, 'NewMexico'), (-1, -4, 'Nevada')] --Sort - ()()()()()
Color Assignment in Progress B -> Idaho
[(-2, -1, 'Montana'), (-2, -3, 'Nevada'), (-1, -2, 'Washington'), (-1, -3, 'Oregon')] ()()()()()
[(-2, -3, 'Nevada'), (-2, -1, 'Montana'), (-1, -3, 'Oregon'), (-1, -2, 'Washington')] --Sort - ()()()()()
Color Assignment in Progress R -> Nevada
[(-1, -3, 'California'), (-3, -2, 'Arizona'), (-2, -2, 'Oregon')] ()()()()()
[(-3, -2, 'Arizona'), (-2, -2, 'Oregon'), (-1, -3, 'California')] --Sort - ()()()()()
Color Assignment in Progress Y -> Arizona
[(-3, -2, 'NewMexico'), (-2, -2, 'California')] ()()()()()
[(-3, -2, 'NewMexico'), (-2, -2, 'California')] --Sort - ()()()()()
Color Assignment in Progress R -> NewMexico
[(-2, -2, 'Oklahoma'), (-1, -2, 'Texas')] ()()()()()
[(-2, -2, 'Oklahoma'), (-1, -2, 'Texas')] --Sort - ()()()()()
Color Assignment in Progress G -> Oklahoma
[(-2, 0, 'Kansas'), (-2, -1, 'Texas')] ()()()()()
[(-2, -1, 'Texas'), (-2, 0, 'Kansas')] --Sort - ()()()()()
Color Assignment in Progress B -> Texas
[(-2, 0, 'Louisiana')] ()()()()()
[(-2, 0, 'Louisiana')] --Sort - ()()()()()
Color Assignment in Progress G -> Louisiana
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Louisiana
Color Assigned B -> Texas
Color Assignment in Progress R -> Kansas
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Kansas
Color Assigned G -> Oklahoma
Color Assigned R -> NewMexico
Color Assignment in Progress B -> California
[(-1, 0, 'Hawaii'), (-2, -1, 'Oregon')] ()()()()()
[(-2, -1, 'Oregon'), (-1, 0, 'Hawaii')] --Sort - ()()()()()
Color Assignment in Progress G -> Oregon
[(-2, -1, 'Washington')] ()()()()()
[(-2, -1, 'Washington')] --Sort - ()()()()()
Color Assignment in Progress R -> Washington
[(-1, 0, 'Alaska')] ()()()()()
[(-1, 0, 'Alaska')] --Sort - ()()()()()
Color Assignment in Progress G -> Alaska
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Alaska
```

```
Color Assigned R -> Washington
Color Assigned G -> Oregon
Color Assignment in Progress R -> Hawaii
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Hawaii
Color Assigned B -> California
Color Assigned Y -> Arizona
Color Assigned R -> Nevada
Color Assignment in Progress G -> Montana
[(-2, -1, 'NorthDakota')] ()()()()()
[(-2, -1, 'NorthDakota')] --Sort - ()()()()()
Color Assignment in Progress R -> NorthDakota
[(-2, -1, 'Minnesota')] ()()()()()
[(-2, -1, 'Minnesota')] --Sort - ()()()()()
Color Assignment in Progress G -> Minnesota
[(-2, -1, 'Wisconsin')] ()()()()()
[(-2, -1, 'Wisconsin')] --Sort - ()()()()()
Color Assignment in Progress B -> Wisconsin
[(-2, -2, 'Michigan')] ()()()()()
[(-2, -2, 'Michigan')] --Sort - ()()()()()
Color Assignment in Progress R -> Michigan
[(-2, -1, 'Indiana'), (-1, -3, 'Ohio')] ()()()()()
[(-2, -1, 'Indiana'), (-1, -3, 'Ohio')] --Sort - ()()()()()
Color Assignment in Progress B -> Indiana
[(-2, -2, 'Ohio')] ()()()()()
[(-2, -2, 'Ohio')] --Sort - ()()()()()
Color Assignment in Progress G -> Ohio
[(-1, -5, 'Pennsylvania'), (-2, -3, 'WestVirginia')] ()()()()()
[(-2, -3, 'WestVirginia'), (-1, -5, 'Pennsylvania')] --Sort - ()()()()()
Color Assignment in Progress B -> WestVirginia
[(-1, -3, 'Maryland'), (-3, -1, 'Virginia'), (-2, -4, 'Pennsylvania')] ()()()()()
[(-3, -1, 'Virginia'), (-2, -4, 'Pennsylvania'), (-1, -3, 'Maryland')] --Sort - ()()()()(
)
Color Assignment in Progress Y -> Virginia
[(-2, -2, 'Maryland')] ()()()()()
[(-2, -2, 'Maryland')] --Sort - ()()()()()
Color Assignment in Progress G -> Maryland
[(-1, -2, 'Delaware'), (-2, -3, 'Pennsylvania')] ()()()()()
[(-2, -3, 'Pennsylvania'), (-1, -2, 'Delaware')] --Sort - ()()()()()
Color Assignment in Progress R -> Pennsylvania
[(-1, -4, 'NewYork'), (-1, -2, 'NewJersey'), (-2, -1, 'Delaware')] ()()()()()
[(-2, -1, 'Delaware'), (-1, -4, 'NewYork'), (-1, -2, 'NewJersey')] --Sort - ()()()()()
Color Assignment in Progress B -> Delaware
[(-2, -1, 'NewJersey')] ()()()()()
[(-2, -1, 'NewJersey')] --Sort - ()()()()()
Color Assignment in Progress G -> NewJersey
[(-2, -3, 'NewYork')] ()()()()()
```

```
[(-2, -3, 'NewYork')] --Sort - ()()()()()
Color Assignment in Progress B -> NewYork
[(-1, -2, 'Connecticut'), (-1, -2, 'Vermont'), (-1, -4, 'Massachusetts')] ()()()()()
[(-1, -4, 'Massachusetts'), (-1, -2, 'Connecticut'), (-1, -2, 'Vermont')] --Sort - ()()()
()()
Color Assignment in Progress R -> Massachusetts
[(-2, -1, 'Connecticut'), (-1, -1, 'RhodeIsland'), (-2, -1, 'Vermont'), (-1, -2, 'NewHamp
shire')] ()()()()()
[(-2, -1, 'Connecticut'), (-2, -1, 'Vermont'), (-1, -2, 'NewHampshire'), (-1, -1, 'RhodeI
sland')] --Sort - ()()()()()
Color Assignment in Progress G -> Connecticut
[(-2, 0, 'RhodeIsland')] ()()()()()
[(-2, 0, 'RhodeIsland')] --Sort - ()()()()()
Color Assignment in Progress B -> RhodeIsland
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned B -> RhodeIsland
Color Assigned G -> Connecticut
Color Assignment in Progress G -> Vermont
[(-2, -1, 'NewHampshire')] ()()()()()
[(-2, -1, 'NewHampshire')] --Sort - ()()()()()
Color Assignment in Progress B -> NewHampshire
[(-1, 0, 'Maine')] ()()()()()
[(-1, 0, 'Maine')] --Sort - ()()()()()
Color Assignment in Progress R -> Maine
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Maine
Color Assigned B -> NewHampshire
Color Assigned G -> Vermont
Color Assigned R -> Massachusetts
Color Assigned B -> NewYork
Color Assigned G -> NewJersey
Color Assigned B -> Delaware
Color Assigned R -> Pennsylvania
Color Assigned G -> Maryland
Color Assigned Y -> Virginia
Color Assigned B -> WestVirginia
Color Assigned G -> Ohio
Color Assigned B -> Indiana
Color Assigned R -> Michigan
Color Assigned B -> Wisconsin
Color Assigned G -> Minnesota
Color Assigned R -> NorthDakota
Color Assigned G -> Montana
Color Assigned B -> Idaho
Color Assigned G -> Utah
Color Assigned B -> Colorado
```

```
Color Assigned R -> Wyoming
Color Assigned B -> SouthDakota
Color Assigned G -> Nebraska
Color Assigned R -> Iowa
Color Assigned G -> Illinois
Color Assigned R -> Kentucky
Color Assigned B -> Missouri
Color Assigned R -> Arkansas
Color Assigned B -> Mississippi
Color Assignment in Progress G -> Florida
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Florida
Color Assigned R -> Alabama
Color Assignment in Progress G -> SouthCarolina
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> SouthCarolina
Color Assigned B -> Georgia
Color Assigned G -> Tennessee
Color Assigned R -> NorthCarolina

Time:  0.4583968460001415
Number of Backtracks 0
```

6. DFS + Heuristic + Singleton

```
1. USA
2. AUS

Select Country:
 1

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
6
Color Assignment in Progress R -> NorthCarolina
[(-1, -1, 4, 'SouthCarolina'), (-1, -4, 4, 'Virginia'), (-1, -7, 4, 'Tennessee'), (-1, -4
, 4, 'Georgia')] ()()()()()
[(-1, -7, 4, 'Tennessee'), (-1, -4, 4, 'Georgia'), (-1, -4, 4, 'Virginia'), (-1, -1, 4, '
SouthCarolina')] --Sort - ()()()()()
Color Assignment in Progress G -> Tennessee
```

```
[(-1, -5, 4, 'Arkansas'), (-1, -7, 4, 'Missouri'), (-1, -6, 4, 'Kentucky'), (-1, -3, 4, '
Mississippi'), (-2, -3, 3, 'Virginia'), (-1, -3, 4, 'Alabama'), (-2, -3, 3, 'Georgia')] (
)()()()()
[(-2, -3, 3, 'Georgia'), (-2, -3, 3, 'Virginia'), (-1, -7, 4, 'Missouri'), (-1, -6, 4, 'K
entucky'), (-1, -5, 4, 'Arkansas'), (-1, -3, 4, 'Alabama'), (-1, -3, 4, 'Mississippi')] -
--Sort - ()()()()()
Color Assignment in Progress B -> Georgia
[(-1, -1, 4, 'Florida'), (-2, 0, 3, 'SouthCarolina'), (-2, -2, 3, 'Alabama')] ()()()()()
[(-2, -2, 3, 'Alabama'), (-2, 0, 3, 'SouthCarolina'), (-1, -1, 4, 'Florida')] --Sort - ()
()()()()
Color Assignment in Progress R -> Alabama
[(-2, 0, 3, 'Florida'), (-2, -2, 3, 'Mississippi')] ()()()()()
[(-2, -2, 3, 'Mississippi'), (-2, 0, 3, 'Florida')] --Sort - ()()()()()
Color Assignment in Progress B -> Mississippi
[(-1, -2, 4, 'Louisiana'), (-2, -4, 3, 'Arkansas')] ()()()()()
[(-2, -4, 3, 'Arkansas'), (-1, -2, 4, 'Louisiana')] --Sort - ()()()()()
Color Assignment in Progress R -> Arkansas
[(-2, -6, 3, 'Missouri'), (-1, -5, 4, 'Oklahoma'), (-1, -3, 4, 'Texas'), (-2, -1, 3, 'Lou
isiana')] ()()()()()
[(-2, -6, 3, 'Missouri'), (-2, -1, 3, 'Louisiana'), (-1, -5, 4, 'Oklahoma'), (-1, -3, 4,
'Texas')] --Sort - ()()()()()
Color Assignment in Progress B -> Missouri
[(-1, -3, 4, 'Kansas'), (-1, -5, 4, 'Iowa'), (-2, -4, 3, 'Oklahoma'), (-1, -5, 4, 'Nebras
ka'), (-1, -5, 4, 'Illinois'), (-2, -5, 3, 'Kentucky')] ()()()()()
[(-2, -5, 3, 'Kentucky'), (-2, -4, 3, 'Oklahoma'), (-1, -5, 4, 'Illinois'), (-1, -5, 4, '
Iowa'), (-1, -5, 4, 'Nebraska'), (-1, -3, 4, 'Kansas')] --Sort - ()()()()()
Color Assignment in Progress R -> Kentucky
[(-2, -4, 3, 'Illinois'), (-1, -4, 4, 'Ohio'), (-1, -3, 4, 'Indiana'), (-2, -2, 2, 'Virgi
nia'), (-1, -4, 4, 'WestVirginia')] ()()()()()
[(-2, -4, 3, 'Illinois'), (-2, -2, 2, 'Virginia'), (-1, -4, 4, 'Ohio'), (-1, -4, 4, 'West
Virginia'), (-1, -3, 4, 'Indiana')] --Sort - ()()()()()
Color Assignment in Progress G -> Illinois
[(-2, -4, 3, 'Iowa'), (-1, -3, 4, 'Wisconsin'), (-1, -3, 4, 'Michigan'), (-2, -2, 3, 'Ind
iana')] ()()()()()
[(-2, -4, 3, 'Iowa'), (-2, -2, 3, 'Indiana'), (-1, -3, 4, 'Michigan'), (-1, -3, 4, 'Wisco
nsin')] --Sort - ()()()()()
Color Assignment in Progress R -> Iowa
[(-2, -4, 3, 'Nebraska'), (-1, -5, 4, 'SouthDakota'), (-2, -2, 3, 'Wisconsin'), (-1, -3,
4, 'Minnesota')] ()()()()()
[(-2, -4, 3, 'Nebraska'), (-2, -2, 3, 'Wisconsin'), (-1, -5, 4, 'SouthDakota'), (-1, -3,
4, 'Minnesota')] --Sort - ()()()()()
Color Assignment in Progress G -> Nebraska
[(-1, -6, 4, 'Colorado'), (-2, -2, 3, 'Kansas'), (-1, -5, 4, 'Wyoming'), (-2, -4, 3, 'Sou
thDakota')] ()()()()()
[(-2, -4, 3, 'SouthDakota'), (-2, -2, 3, 'Kansas'), (-1, -6, 4, 'Colorado'), (-1, -5, 4,
'Wyoming')] --Sort - ()()()()()
Color Assignment in Progress B -> SouthDakota
```

```
[(-1, -3, 4, 'Montana'), (-2, -4, 3, 'Wyoming'), (-1, -2, 4, 'NorthDakota'), (-2, -2, 3,
'Minnesota')] ()()()()()
[(-2, -4, 3, 'Wyoming'), (-2, -2, 3, 'Minnesota'), (-1, -3, 4, 'Montana'), (-1, -2, 4, 'N
orthDakota')] --Sort - ()()()()()
Color Assignment in Progress R -> Wyoming
[(-2, -5, 3, 'Colorado'), (-2, -2, 3, 'Montana'), (-1, -5, 4, 'Idaho'), (-1, -5, 4, 'Utah
')] ()()()()()
[(-2, -5, 3, 'Colorado'), (-2, -2, 3, 'Montana'), (-1, -5, 4, 'Idaho'), (-1, -5, 4, 'Utah
')] --Sort - ()()()()()
Color Assignment in Progress B -> Colorado
[(-1, -4, 4, 'NewMexico'), (-2, -1, 2, 'Kansas'), (-2, -3, 2, 'Oklahoma'), (-1, -4, 4, 'A
rizona'), (-2, -4, 3, 'Utah')] ()()()()()
[(-2, -4, 3, 'Utah'), (-2, -3, 2, 'Oklahoma'), (-2, -1, 2, 'Kansas'), (-1, -4, 4, 'Arizon
a'), (-1, -4, 4, 'NewMexico')] --Sort - ()()()()()
Color Assignment in Progress G -> Utah
[(-2, -3, 3, 'NewMexico'), (-1, -4, 4, 'Nevada'), (-2, -3, 3, 'Arizona'), (-2, -4, 3, 'Id
aho')] ()()()()()
[(-2, -4, 3, 'Idaho'), (-2, -3, 3, 'Arizona'), (-2, -3, 3, 'NewMexico'), (-1, -4, 4, 'Nev
ada')] --Sort - ()()()()()
Color Assignment in Progress B -> Idaho
[(-2, -1, 2, 'Montana'), (-2, -3, 3, 'Nevada'), (-1, -2, 4, 'Washington'), (-1, -3, 4, 'O
regon')] ()()()()()
[(-2, -3, 3, 'Nevada'), (-2, -1, 2, 'Montana'), (-1, -3, 4, 'Oregon'), (-1, -2, 4, 'Washi
ngton')] --Sort - ()()()()()
Color Assignment in Progress R -> Nevada
[(-1, -3, 4, 'California'), (-3, -2, 2, 'Arizona'), (-2, -2, 3, 'Oregon')] ()()()()()
[(-3, -2, 2, 'Arizona'), (-2, -2, 3, 'Oregon'), (-1, -3, 4, 'California')] --Sort - ()()(
)()()
Color Assignment in Progress Y -> Arizona
[(-3, -2, 2, 'NewMexico'), (-2, -2, 3, 'California')] ()()()()()
[(-3, -2, 2, 'NewMexico'), (-2, -2, 3, 'California')] --Sort - ()()()()()
Color Assignment in Progress R -> NewMexico
[(-2, -2, 2, 'Oklahoma'), (-1, -2, 3, 'Texas')] ()()()()()
[(-2, -2, 2, 'Oklahoma'), (-1, -2, 3, 'Texas')] --Sort - ()()()()()
Color Assignment in Progress G -> Oklahoma
[(-2, 0, 2, 'Kansas'), (-2, -1, 3, 'Texas')] ()()()()()
[(-2, -1, 3, 'Texas'), (-2, 0, 2, 'Kansas')] --Sort - ()()()()()
Color Assignment in Progress B -> Texas
[(-2, 0, 2, 'Louisiana')] ()()()()()
[(-2, 0, 2, 'Louisiana')] --Sort - ()()()()()
Color Assignment in Progress G -> Louisiana
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Louisiana
Color Assigned B -> Texas
Color Assignment in Progress R -> Kansas
[] ()()()()()
[] --Sort - ()()()()()
```

```
Color Assigned R -> Kansas
Color Assigned G -> Oklahoma
Color Assigned R -> NewMexico
Color Assignment in Progress B -> California
[(-1, 0, 4, 'Hawaii'), (-2, -1, 2, 'Oregon')] ()()()()()
[(-2, -1, 2, 'Oregon'), (-1, 0, 4, 'Hawaii')] --Sort - ()()()()()
Color Assignment in Progress G -> Oregon
[(-2, -1, 3, 'Washington')] ()()()()()
[(-2, -1, 3, 'Washington')] --Sort - ()()()()()
Color Assignment in Progress R -> Washington
[(-1, 0, 4, 'Alaska')] ()()()()()
[(-1, 0, 4, 'Alaska')] --Sort - ()()()()()
Color Assignment in Progress G -> Alaska
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Alaska
Color Assigned R -> Washington
Color Assigned G -> Oregon
Color Assignment in Progress R -> Hawaii
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Hawaii
Color Assigned B -> California
Color Assigned Y -> Arizona
Color Assigned R -> Nevada
Color Assignment in Progress G -> Montana
[(-2, -1, 3, 'NorthDakota')] ()()()()()
[(-2, -1, 3, 'NorthDakota')] --Sort - ()()()()()
Color Assignment in Progress R -> NorthDakota
[(-2, -1, 2, 'Minnesota')] ()()()()()
[(-2, -1, 2, 'Minnesota')] --Sort - ()()()()()
Color Assignment in Progress G -> Minnesota
[(-2, -1, 2, 'Wisconsin')] ()()()()()
[(-2, -1, 2, 'Wisconsin')] --Sort - ()()()()()
Color Assignment in Progress B -> Wisconsin
[(-2, -2, 3, 'Michigan')] ()()()()()
[(-2, -2, 3, 'Michigan')] --Sort - ()()()()()
Color Assignment in Progress R -> Michigan
[(-2, -1, 2, 'Indiana'), (-1, -3, 3, 'Ohio')] ()()()()()
[(-2, -1, 2, 'Indiana'), (-1, -3, 3, 'Ohio')] --Sort - ()()()()()
Color Assignment in Progress B -> Indiana
[(-2, -2, 3, 'Ohio')] ()()()()()
[(-2, -2, 3, 'Ohio')] --Sort - ()()()()()
Color Assignment in Progress G -> Ohio
[(-1, -5, 4, 'Pennsylvania'), (-2, -3, 3, 'WestVirginia')] ()()()()()
[(-2, -3, 3, 'WestVirginia'), (-1, -5, 4, 'Pennsylvania')] --Sort - ()()()()()
Color Assignment in Progress B -> WestVirginia
```

```
[(-1, -3, 4, 'Maryland'), (-3, -1, 2, 'Virginia'), (-2, -4, 3, 'Pennsylvania')] ()()()()(
)
[(-3, -1, 2, 'Virginia'), (-2, -4, 3, 'Pennsylvania'), (-1, -3, 4, 'Maryland')] --Sort -
()()()()()
Color Assignment in Progress Y -> Virginia
[(-2, -2, 3, 'Maryland')] ()()()()()
[(-2, -2, 3, 'Maryland')] --Sort - ()()()()()
Color Assignment in Progress G -> Maryland
[(-1, -2, 4, 'Delaware'), (-2, -3, 2, 'Pennsylvania')] ()()()()()
[(-2, -3, 2, 'Pennsylvania'), (-1, -2, 4, 'Delaware')] --Sort - ()()()()()
Color Assignment in Progress R -> Pennsylvania
[(-1, -4, 4, 'NewYork'), (-1, -2, 4, 'NewJersey'), (-2, -1, 3, 'Delaware')] ()()()()()
[(-2, -1, 3, 'Delaware'), (-1, -4, 4, 'NewYork'), (-1, -2, 4, 'NewJersey')] --Sort - ()()
()()()
Color Assignment in Progress B -> Delaware
[(-2, -1, 3, 'NewJersey')] ()()()()()
[(-2, -1, 3, 'NewJersey')] --Sort - ()()()()()
Color Assignment in Progress G -> NewJersey
[(-2, -3, 3, 'NewYork')] ()()()()()
[(-2, -3, 3, 'NewYork')] --Sort - ()()()()()
Color Assignment in Progress B -> NewYork
[(-1, -2, 4, 'Connecticut'), (-1, -2, 4, 'Vermont'), (-1, -4, 4, 'Massachusetts')] ()()()
()()
[(-1, -4, 4, 'Massachusetts'), (-1, -2, 4, 'Connecticut'), (-1, -2, 4, 'Vermont')] --Sort
- ()()()()()
Color Assignment in Progress R -> Massachusetts
[(-2, -1, 3, 'Connecticut'), (-1, -1, 4, 'RhodeIsland'), (-2, -1, 3, 'Vermont'), (-1, -2,
4, 'NewHampshire')] ()()()()()
[(-2, -1, 3, 'Connecticut'), (-2, -1, 3, 'Vermont'), (-1, -2, 4, 'NewHampshire'), (-1, -1
, 4, 'RhodeIsland')] --Sort - ()()()()()
Color Assignment in Progress G -> Connecticut
[(-2, 0, 3, 'RhodeIsland')] ()()()()()
[(-2, 0, 3, 'RhodeIsland')] --Sort - ()()()()()
Color Assignment in Progress B -> RhodeIsland
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned B -> RhodeIsland
Color Assigned G -> Connecticut
Color Assignment in Progress G -> Vermont
[(-2, -1, 3, 'NewHampshire')] ()()()()()
[(-2, -1, 3, 'NewHampshire')] --Sort - ()()()()()
Color Assignment in Progress B -> NewHampshire
[(-1, 0, 4, 'Maine')] ()()()()()
[(-1, 0, 4, 'Maine')] --Sort - ()()()()()
Color Assignment in Progress R -> Maine
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Maine
```

```
Color Assigned B -> NewHampshire
Color Assigned G -> Vermont
Color Assigned R -> Massachusetts
Color Assigned B -> NewYork
Color Assigned G -> NewJersey
Color Assigned B -> Delaware
Color Assigned R -> Pennsylvania
Color Assigned G -> Maryland
Color Assigned Y -> Virginia
Color Assigned B -> WestVirginia
Color Assigned G -> Ohio
Color Assigned B -> Indiana
Color Assigned R -> Michigan
Color Assigned B -> Wisconsin
Color Assigned G -> Minnesota
Color Assigned R -> NorthDakota
Color Assigned G -> Montana
Color Assigned B -> Idaho
Color Assigned G -> Utah
Color Assigned B -> Colorado
Color Assigned R -> Wyoming
Color Assigned B -> SouthDakota
Color Assigned G -> Nebraska
Color Assigned R -> Iowa
Color Assigned G -> Illinois
Color Assigned R -> Kentucky
Color Assigned B -> Missouri
Color Assigned R -> Arkansas
Color Assigned B -> Mississippi
Color Assignment in Progress G -> Florida
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> Florida
Color Assigned R -> Alabama
Color Assignment in Progress G -> SouthCarolina
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned G -> SouthCarolina
Color Assigned B -> Georgia
Color Assigned G -> Tennessee
Color Assigned R -> NorthCarolina


Time:  0.5130860879999091
Number of Backtracks 0
```

For Australia

1. DFS

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
1
Color Assigned G -> WestAustralia
Color Assigned R -> NorthTerritory
Color Assigned G -> Queensland
Color Assigned B -> SouthAustralia
Color Assigned R -> Tasmania
Color Assigned G -> Victoria
Color Assigned R -> NewSouthWales

Time:  0.0013495219991455087
Number of Backtracks 0
```

2. DFS + FC

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
2
Color Assignment in Progress R -> NewSouthWales
Color Assignment in Progress G -> Victoria
Color Assignment in Progress B -> SouthAustralia
Color Assignment in Progress G -> Queensland
Color Assignment in Progress R -> NorthTerritory
Color Assignment in Progress G -> WestAustralia
Color Assigned G -> WestAustralia
Color Assigned R -> NorthTerritory
Color Assigned G -> Queensland
Color Assigned B -> SouthAustralia
Color Assignment in Progress R -> Tasmania
```

```
Color Assigned R -> Tasmania
Color Assigned G -> Victoria
Color Assigned R -> NewSouthWales

Time:  0.001243454000359634
Number of Backtracks 0
```

3. DFS + FC + Singleton

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
3
Color Assignment in Progress R -> NewSouthWales
Color Assignment in Progress G -> Victoria
Color Assignment in Progress B -> SouthAustralia
Color Assignment in Progress G -> Queensland
Color Assignment in Progress R -> NorthTerritory
Color Assignment in Progress G -> WestAustralia
Color Assigned G -> WestAustralia
Color Assigned R -> NorthTerritory
Color Assigned G -> Queensland
Color Assigned B -> SouthAustralia
Color Assignment in Progress R -> Tasmania
Color Assigned R -> Tasmania
Color Assigned G -> Victoria
Color Assigned R -> NewSouthWales

Time:  0.004226635000122769
Number of Backtracks 0
```

4. DFS + Heuristic

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
4
[(-1, -2, 'Victoria'), (-1, -2, 'Queensland'), (-1, -4, 'SouthAustralia')] ()()()()()
```

```
[(-1, -4, 'SouthAustralia'), (-1, -2, 'Queensland'), (-1, -2, 'Victoria')] --Sort - (
)()()()()
[(-2, -1, 'Queensland'), (-1, -1, 'WestAustralia'), (-1, -2, 'NorthTerritory'), (-2,
-1, 'Victoria')] ()()()()()
[(-2, -1, 'Queensland'), (-2, -1, 'Victoria'), (-1, -2, 'NorthTerritory'), (-1, -1, '
WestAustralia')] --Sort - ()()()()()
[(-2, -1, 'NorthTerritory')] ()()()()()
[(-2, -1, 'NorthTerritory')] --Sort - ()()()()()
[(-2, 0, 'WestAustralia')] ()()()()()
[(-2, 0, 'WestAustralia')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned B -> WestAustralia
Color Assigned R -> NorthTerritory
Color Assigned B -> Queensland
[(-1, 0, 'Tasmania')] ()()()()()
[(-1, 0, 'Tasmania')] --Sort - ()()()()()
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Tasmania
Color Assigned B -> Victoria
Color Assigned G -> SouthAustralia
Color Assigned R -> NewSouthWales

Time:  0.001831579999816313
Number of Backtracks 0
```

5. DFS + Heuristic + FC

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
5
Color Assignment in Progress R -> NewSouthWales
[(-1, -2, 'Victoria'), (-1, -2, 'Queensland'), (-1, -4, 'SouthAustralia')] ()()()()()
[(-1, -4, 'SouthAustralia'), (-1, -2, 'Queensland'), (-1, -2, 'Victoria')] --Sort - (
)()()()()
Color Assignment in Progress G -> SouthAustralia
[(-2, -1, 'Queensland'), (-1, -1, 'WestAustralia'), (-1, -2, 'NorthTerritory'), (-2,
-1, 'Victoria')] ()()()()()
[(-2, -1, 'Queensland'), (-2, -1, 'Victoria'), (-1, -2, 'NorthTerritory'), (-1, -1, '
WestAustralia')] --Sort - ()()()()()
Color Assignment in Progress B -> Queensland
[(-2, -1, 'NorthTerritory')] ()()()()()
[(-2, -1, 'NorthTerritory')] --Sort - ()()()()()
Color Assignment in Progress R -> NorthTerritory
[(-2, 0, 'WestAustralia')] ()()()()()
[(-2, 0, 'WestAustralia')] --Sort - ()()()()()
```

```
Color Assignment in Progress B -> WestAustralia
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned B -> WestAustralia
Color Assigned R -> NorthTerritory
Color Assigned B -> Queensland
Color Assignment in Progress B -> Victoria
[(-1, 0, 'Tasmania')] ()()()()()
[(-1, 0, 'Tasmania')] --Sort - ()()()()()
Color Assignment in Progress R -> Tasmania
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Tasmania
Color Assigned B -> Victoria
Color Assigned G -> SouthAustralia
Color Assigned R -> NewSouthWales

Time:  0.0029856450000806944
Number of Backtracks 0
```

6. DFS + Heuristic, FC + Singleton

```
1. USA
2. AUS

Select Country:
 2

1. DFS
2. DFS + FC
3. DFS + FC + Singleton
4. DFS + Heuristic
5. DFS + Heuristic + FC
6. DFS + heuristic, FC + Singleton

Choose the Algorithm:
6
Color Assignment in Progress R -> NewSouthWales
[(-1, -2, 3, 'Victoria'), (-1, -2, 3, 'Queensland'), (-1, -4, 3, 'SouthAustralia')] (
)()()()()
[(-1, -4, 3, 'SouthAustralia'), (-1, -2, 3, 'Queensland'), (-1, -2, 3, 'Victoria')] -
-Sort - ()()()()()
Color Assignment in Progress G -> SouthAustralia
[(-2, -1, 2, 'Queensland'), (-1, -1, 3, 'WestAustralia'), (-1, -2, 3, 'NorthTerritory
'), (-2, -1, 2, 'Victoria')] ()()()()()
[(-2, -1, 2, 'Queensland'), (-2, -1, 2, 'Victoria'), (-1, -2, 3, 'NorthTerritory'), (
-1, -1, 3, 'WestAustralia')] --Sort - ()()()()()
Color Assignment in Progress B -> Queensland
[(-2, -1, 2, 'NorthTerritory')] ()()()()()
[(-2, -1, 2, 'NorthTerritory')] --Sort - ()()()()()
Color Assignment in Progress R -> NorthTerritory
[(-2, 0, 2, 'WestAustralia')] ()()()()()
[(-2, 0, 2, 'WestAustralia')] --Sort - ()()()()()
Color Assignment in Progress B -> WestAustralia
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned B -> WestAustralia
Color Assigned R -> NorthTerritory
```

```
Color Assigned B -> Queensland
Color Assignment in Progress B -> Victoria
[(-1, 0, 3, 'Tasmania')] ()()()()()
[(-1, 0, 3, 'Tasmania')] --Sort - ()()()()()
Color Assignment in Progress R -> Tasmania
[] ()()()()()
[] --Sort - ()()()()()
Color Assigned R -> Tasmania
Color Assigned B -> Victoria
Color Assigned G -> SouthAustralia
Color Assigned R -> NewSouthWales

Time:  0.0023720449999018456
Number of Backtracks 0
```

OUTPUT TABLE

Country	With Heuristics	DFS	DFS+FC	DFS+FC+Singleton	Backtrack	Time
AUSTRALIA	Yes	Yes	No	No	0	0.001831579999816
AUSTRALIA	Yes	Yes	Yes	No	0	0.00298564500008
AUSTRALIA	Yes	Yes	Yes	Yes	0	0.002372044999901845
AUSTRALIA	No	Yes	No	No	0	0.001349521999145508
AUSTRALIA	No	Yes	Yes	No	0	0. 00124345400035963
AUSTRALIA	No	Yes	Yes	Yes	0	0. 00422663500012276
USA	Yes	Yes	No	No	0	0.4287903370004642
USA	Yes	Yes	Yes	No	0	0.4583968460001415
USA	Yes	Yes	Yes	Yes	0	0.5130860879999091
USA	No	Yes	No	No	73	0.37268043200037937
USA	No	Yes	Yes	No	15	0.428951492999658
USA	No	Yes	Yes	Yes	0	0.41947213700041175

SOURCE CODE

```
import matplotlib.pyplot as plt

import matplotlib.pyplot as plt1

from mpl_toolkits.basemap import Basemap
```

```
from matplotlib.patches import Polygon
import networkx as nx
import copy
import webbrowser
import timeit


paint_true = {}
backtrack = 0
if_singleton = 0
heuristic = 0


basemap = Basemap(llcrnrlon=-119, llcrnrlat=22, urcrnrlon=-64, urcrnrlat=49, projection='lcc',
lat_1=33, lat_2=45,
            lon_0=-95)


basemap.readshapefile(r'''/Users/sriganeshlokesh/anaconda3/lib/python3.7/site-
packages/mpl_toolkits/basemap/st99_d00''', name='states', drawbounds=True)
states = []
for s_dict in basemap.states_info:
   states.append(s_dict['NAME'])
ax = plt.gca()



def check(map):
   for c, a in map.items():
      assert (c not in a)
      for l in a:
         assert (l in map and c in map[l])
```

```
def value_next(r, n, assignment):
    if heuristic == 0:
        return n[r]
    else:
        if if_singleton == 0:
            inf = [
                (
                    -len({paint_true[next] for next in n[number] if next in paint_true}),

                    -len({next for next in n[number] if next not in paint_true}),

                    number
                ) for number in n[r] if number not in paint_true]
        else:
            inf = [
                (
                    -len({paint_true[next] for next in n[number] if next in paint_true}),

                    -len({next for next in n[number] if next not in paint_true}),

                    len(assignment[number]),
                    number
                ) for number in n[r] if number not in paint_true]

        print(inf, "()()()()()")
        inf.sort()
```

```
        print(inf, "--Sort - ()()()()()")
        if if_singleton == 0:
            candidates = [number for _, _, number in inf]
        else:
            candidates = [number for _, _, _, number in inf]
        return candidates



def color_assign(r, n, assignment):
    if heuristic == 0:
        return assignment[r]
    else:
        a = []
        for s_color in assignment[r]:
            c_total = 0
            a.append([s_color])
            for next in n[r]:
                if s_color in assignment[next]:
                    c_total = c_total + len(assignment[next]) - 1
                else:
                    c_total = c_total + len(assignment[next])
            a[a.index([s_color])].append(c_total)
        a = sorted(a, key=lambda a_sort: a_sort[1], reverse=True)
        a = [a_sort[0] for a_sort in a]
        return a
```

```
def dfs(r, n, assignment):
    add_color = 0
    w = 0
    global backtrack
    for s_color in color_assign(r, n, assignment):
        for j in n[r]:
            if j in paint_true and paint_true[j] == s_color:
                add_color = 1
                break
        if add_color == 1:
            add_color = 0
            continue
        paint_true[r] = s_color
        for k in value_next(r, n, assignment):
            if k not in paint_true:
                if (dfs(k, n, assignment) == False):
                    paint_true.pop(r)
                    w = 1
                    break
        if w == 0:
            print("Color Assigned %s -> %s" % (paint_true[r], r))
            return True
        else:
            w = 0
            continue
    backtrack = backtrack + 1
    return False
```

```
def decrease(r, n, a_c):
    for j in n[r]:
        if paint_true[r] in a_c[j]:
            a_c[j].remove(paint_true[r])



def decrease_for_forward_check(s_color, r, n, a_c):
    copy_a = copy.deepcopy(a_c)
    for j in n[r]:
        if s_color in copy_a[j]:
            copy_a[j].remove(s_color)
        if not d_validate(j, copy_a):
            return False
    return True



def d_validate(r, a_c):
    if not (a_c[r]):
        return False
    return True



def dfs_forward(r, n, assignment):
    w = 0
    d = copy.deepcopy(assignment)
```

```
global backtrack
for s_color in color_assign(r, n, assignment):


    a = copy.deepcopy(d)
    if decrease_for_forward_check(s_color, r, n, a) == False:
        continue
    paint_true[r] = s_color
    print("Color Assignment in Progress %s -> %s" % (s_color, r))
    decrease(r, n, a)
    a[r] = s_color
    if if_singleton == 1 and heuristic == 0:
        n[r] = sorted(n[r], key=lambda a_sort: len(assignment[a_sort]),
                              reverse=False)
    for next in value_next(r, n, assignment):
        if next not in paint_true:
            if (dfs_forward(next, n, a)) == False:
                paint_true.pop(r)
                w = 1
                break
    if w == 0:
        print("Color Assigned %s -> %s" % (paint_true[r], r))
        return True
    else:
        w = 0
        continue
backtrack = backtrack + 1
return False
```

```
WestAustralia = 'WestAustralia'
NorthTerritory = 'NorthTerritory'
SouthAustralia = 'SouthAustralia'
Queensland = 'Queensland'
NewSouthWales = 'NewSouthWales'
Victoria = 'Victoria'
Tasmania = 'Tasmania'

Aussie = {
   Tasmania: {Victoria},
    WestAustralia: {NorthTerritory, SouthAustralia},
    NorthTerritory: {WestAustralia, Queensland, SouthAustralia},
    SouthAustralia: {WestAustralia, NorthTerritory, Queensland, NewSouthWales, Victoria},
    Queensland: {NorthTerritory, SouthAustralia, NewSouthWales},
    NewSouthWales: {Queensland, SouthAustralia, Victoria},
    Victoria: {SouthAustralia, NewSouthWales, Tasmania}
}

aussie_color = {
    Tasmania: ['R', 'G', 'B'],
    WestAustralia: ['R', 'G', 'B'],
    NorthTerritory: ['R', 'G', 'B'],
    SouthAustralia: ['R', 'G', 'B'],
    Queensland: ['R', 'G', 'B'],
    NewSouthWales: ['R', 'G', 'B'],
    Victoria: ['R', 'G', 'B']
```

}

Alabama = "Alabama"

Alaska = "Alaska"

Arizona = "Arizona"

Arkansas = "Arkansas"

California = "California"

Colorado = "Colorado"

Connecticut = "Connecticut"

Delaware = "Delaware"

Florida = "Florida"

Georgia = "Georgia"

Hawaii = "Hawaii"

Idaho = "Idaho"

Illinois = "Illinois"

Indiana = "Indiana"

Iowa = "Iowa"

Kansas = "Kansas"

Kentucky = "Kentucky"

Louisiana = "Louisiana"

Maine = "Maine"

Maryland = "Maryland"

Massachusetts = "Massachusetts"

Michigan = "Michigan"

Minnesota = "Minnesota"

Mississippi = "Mississippi"

Missouri = "Missouri"

Montana = "Montana"

Nebraska = "Nebraska"

Nevada = "Nevada"

NewHampshire = "NewHampshire"

NewJersey = "NewJersey"

NewMexico = "NewMexico"

NewYork = "NewYork"

NorthCarolina = "NorthCarolina"

NorthDakota = "NorthDakota"

Ohio = "Ohio"

Oklahoma = "Oklahoma"

Oregon = "Oregon"

Pennsylvania = "Pennsylvania"

RhodeIsland = "RhodeIsland"

SouthCarolina = "SouthCarolina"

SouthDakota = "SouthDakota"

Tennessee = "Tennessee"

Texas = "Texas"

Utah = "Utah"

Virginia = "Virginia"

Vermont = "Vermont"

Washington = "Washington"

WestVirginia = "WestVirginia"

Wisconsin = "Wisconsin"

Wyoming = "Wyoming"

USA_S = {

Alabama: {Georgia, Florida, Tennessee, Mississippi},

Alaska: {Washington},

Arizona: {California, Nevada, Utah, Colorado, NewMexico},

Arkansas: {Missouri, Oklahoma, Texas, Louisiana, Tennessee, Mississippi},

California: {Oregon, Nevada, Arizona, Hawaii},

Colorado: {Wyoming, Nebraska, Kansas, Oklahoma, NewMexico, Arizona, Utah},

Connecticut: {NewYork, RhodeIsland, Massachusetts},

Delaware: {Maryland, Pennsylvania, NewJersey},

Florida: {Alabama, Georgia},

Georgia: {SouthCarolina, NorthCarolina, Tennessee, Alabama, Florida},

Hawaii: {California},

Idaho: {Washington, Montana, Oregon, Wyoming, Utah, Nevada},

Illinois: {Wisconsin, Iowa, Missouri, Kentucky, Indiana, Michigan},

Indiana: {Michigan, Illinois, Kentucky, Ohio},

Iowa: {Minnesota, SouthDakota, Nebraska, Missouri, Wisconsin, Illinois},

Kansas: {Nebraska, Colorado, Oklahoma, Missouri},

Kentucky: {Indiana, Illinois, Missouri, Tennessee, Ohio, WestVirginia, Virginia},

Louisiana: {Arkansas, Texas, Mississippi},

Maine: {NewHampshire},

Maryland: {Pennsylvania, WestVirginia, Virginia, Delaware},

Massachusetts: {NewYork, Vermont, NewHampshire, Connecticut, RhodeIsland},

Michigan: {Illinois, Wisconsin, Indiana, Ohio},

Minnesota: {NorthDakota, SouthDakota, Iowa, Wisconsin},

Mississippi: {Tennessee, Arkansas, Louisiana, Alabama},

Missouri: {Iowa, Nebraska, Kansas, Oklahoma, Arkansas, Illinois, Kentucky, Tennessee},

Montana: {Idaho, Wyoming, SouthDakota, NorthDakota},

Nebraska: {SouthDakota, Colorado, Wyoming, Kansas, Missouri, Iowa},

```
    Nevada: {Oregon, Idaho, Utah, Arizona, California},
    NewHampshire: {Maine, Vermont, Massachusetts},
    NewJersey: {NewYork, Pennsylvania, Delaware},
    NewMexico: {Arizona, Utah, Colorado, Oklahoma, Texas},
    NewYork: {Pennsylvania, NewJersey, Connecticut, Massachusetts, Vermont},
    NorthCarolina: {Georgia, Tennessee, SouthCarolina, Virginia},
    NorthDakota: {Montana, SouthDakota, Minnesota},
    Ohio: {Michigan, Indiana, Kentucky, WestVirginia, Pennsylvania},
    Oklahoma: {Kansas, Colorado, NewMexico, Texas, Arkansas, Missouri},
    Oregon: {Washington, Idaho, Nevada, California},
    Pennsylvania: {Ohio, WestVirginia, Delaware, NewJersey, NewYork, Maryland},
    RhodeIsland: {Connecticut, Massachusetts},
    SouthCarolina: {Georgia, NorthCarolina},
    SouthDakota: {NorthDakota, Montana, Wyoming, Nebraska, Minnesota, Iowa},
    Tennessee: {Kentucky, Arkansas, Mississippi, Missouri, Alabama, Georgia, NorthCarolina,
Virginia},
    Texas: {Oklahoma, NewMexico, Arkansas, Louisiana},
    Utah: {Idaho, Nevada, Wyoming, Colorado, Arizona, NewMexico},
    Vermont: {Massachusetts, NewYork, NewHampshire},
    Virginia: {WestVirginia, Kentucky, NorthCarolina, Tennessee, Maryland},
    Washington: {Oregon, Idaho, Alaska},
    WestVirginia: {Ohio, Virginia, Kentucky, Pennsylvania, Maryland},
    Wisconsin: {Minnesota, Illinois, Michigan, Iowa},
    Wyoming: {Montana, SouthDakota, Nebraska, Colorado, Utah, Idaho},
}

US_colors = {
```

Alabama: ['R', 'G', 'B', 'Y'],

Alaska: ['R', 'G', 'B', 'Y'],

Arizona: ['R', 'G', 'B', 'Y'],

Arkansas: ['R', 'G', 'B', 'Y'],

California: ['R', 'G', 'B', 'Y'],

Colorado: ['R', 'G', 'B', 'Y'],

Connecticut: ['R', 'G', 'B', 'Y'],

Delaware: ['R', 'G', 'B', 'Y'],

Florida: ['R', 'G', 'B', 'Y'],

Georgia: ['R', 'G', 'B', 'Y'],

Hawaii: ['R', 'G', 'B', 'Y'],

Idaho: ['R', 'G', 'B', 'Y'],

Illinois: ['R', 'G', 'B', 'Y'],

Indiana: ['R', 'G', 'B', 'Y'],

Iowa: ['R', 'G', 'B', 'Y'],

Kansas: ['R', 'G', 'B', 'Y'],

Kentucky: ['R', 'G', 'B', 'Y'],

Louisiana: ['R', 'G', 'B', 'Y'],

Maine: ['R', 'G', 'B', 'Y'],

Maryland: ['R', 'G', 'B', 'Y'],

Massachusetts: ['R', 'G', 'B', 'Y'],

Michigan: ['R', 'G', 'B', 'Y'],

Minnesota: ['R', 'G', 'B', 'Y'],

Mississippi: ['R', 'G', 'B', 'Y'],

Missouri: ['R', 'G', 'B', 'Y'],

Montana: ['R', 'G', 'B', 'Y'],

Nebraska: ['R', 'G', 'B', 'Y'],

```
    Nevada: ['R', 'G', 'B', 'Y'],
    NewHampshire: ['R', 'G', 'B', 'Y'],
    NewJersey: ['R', 'G', 'B', 'Y'],
    NewMexico: ['R', 'G', 'B', 'Y'],
    NewYork: ['R', 'G', 'B', 'Y'],
    NorthCarolina: ['R', 'G', 'B', 'Y'],
    NorthDakota: ['R', 'G', 'B', 'Y'],
    Ohio: ['R', 'G', 'B', 'Y'],
    Oklahoma: ['R', 'G', 'B', 'Y'],
    Oregon: ['R', 'G', 'B', 'Y'],
    Pennsylvania: ['R', 'G', 'B', 'Y'],
    RhodeIsland: ['R', 'G', 'B', 'Y'],
    SouthCarolina: ['R', 'G', 'B', 'Y'],
    SouthDakota: ['R', 'G', 'B', 'Y'],
    Tennessee: ['R', 'G', 'B', 'Y'],
    Texas: ['R', 'G', 'B', 'Y'],
    Utah: ['R', 'G', 'B', 'Y'],
    Virginia: ['R', 'G', 'B', 'Y'],
    Vermont: ['R', 'G', 'B', 'Y'],
    Washington: ['R', 'G', 'B', 'Y'],
    WestVirginia: ['R', 'G', 'B', 'Y'],
    Wisconsin: ['R', 'G', 'B', 'Y'],
    Wyoming: ['R', 'G', 'B', 'Y'],
}
USA_S = {number: next for number, next in USA_S.items() if next}
def build_graph():
    g = nx.Graph()
```

```
    for e in tempList:
        g.add(e[0], e[1])
    return g



def paint_graph(G, paint_true):
    pos = nx.spring_layout(G)
    val = paint_true.values()
    nx.draw(G, pos, with_labels=True, node_size=500, node_color=val, edge_color='black', width=1,
            alpha=.7)
if __name__ == '__main__':
    print("\n1. USA")
    print("2. AUS")
    n_name = int(input("\nSelect Country:\n "))
    country = ""
    complete = {}
    s_color = {}
    short = ""

    if n_name == 1:
        country = "USA"
        complete = USA_S
        s_color = US_colors
        short = NorthCarolina
        flag = 1
    elif n_name == 2:
        country = "AUS"
```

```
    complete = Aussie
    s_color = aussie_color
    short = NSW
    flag = 2
else:
    print("Invalid!, Enter Valid Value")
    exit(0)

check(USA_S)

print("\n1. DFS")
print("2. DFS + FC")
print("3. DFS + FC + Singleton")
print("4. DFS + Heuristic")
print("5. DFS + Heuristic + FC")
print("6. DFS + heuristic, FC + Singleton")

algo = int(input("\nChoose the Algorithm:\n"))
start = timeit.default_timer()
if algo == 1:
    if (dfs(short, complete, s_color)):
        stt = list(paint_true.keys())
        clr = list(paint_true.values())
        if flag == 1:
            for i in range(len(paint_true.keys())):
                seg = basemap.states[states.index(stt[i])]
                p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
```

```
            ax.add_patch(p)
        plt.show()


elif algo == 2:
    if (dfs_forward(short, complete, s_color)):
        stt = list(paint_true.keys())
        clr = list(paint_true.values())
        if flag == 1:
            for i in range(len(paint_true.keys())):
                seg = basemap.states[states.index(stt[i])]
                p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
                ax.add_patch(p)
            plt.show()
elif algo == 3:
    if_singleton = 1
    if dfs_forward(short, complete, s_color):
        stt = list(paint_true.keys())
        clr = list(paint_true.values())
        if flag == 1:
            for i in range(len(paint_true.keys())):
                seg = basemap.states[states.index(stt[i])]
                p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
                ax.add_patch(p)
            plt.show()
elif algo == 4:
    heuristic = 1
    if (dfs(short, complete, s_color)):
```

```
        stt = list(paint_true.keys())
        clr = list(paint_true.values())
        if flag == 1:
            for i in range(len(paint_true.keys())):
                seg = basemap.states[states.index(stt[i])]
                p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
                ax.add_patch(p)
            plt.show()
    elif algo == 5:
        heuristic = 1
        if (dfs_forward(short, complete, s_color)):
            stt = list(paint_true.keys())
            clr = list(paint_true.values())
            if flag == 1:
                for i in range(len(paint_true.keys())):
                    seg = basemap.states[states.index(stt[i])]
                    p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
                    ax.add_patch(p)
                plt.show()
    elif algo == 6:
        heuristic = 1
        if_singleton = 1
        if (dfs_forward(short, complete, s_color)):
            stt = list(paint_true.keys())
            clr = list(paint_true.values())
            if flag == 1:
                for i in range(len(paint_true.keys())):
```

```
            seg = basemap.states[states.index(stt[i])]
            p = Polygon(seg, facecolor=(clr[i]), edgecolor=clr[i])
            ax.add_patch(p)
        plt.show()
else:
    print("Invalid Value, Enter Valid Value")
    exit(0)
timer_stop = timeit.default_timer()
print('\nTime: ', timer_stop - start)
print("Number of Backtracks", backtrack)
paint_true.clear()
```

CITATION

- Map Coloring a Problem, Geeks for Geeks.
- Map Coloring Problem, Wikipedia.