

Ex. No: 04

Constraint Satisfaction Problem and Map Coloring

09/09/2022

Code:

class Continent:

```
def __init__(self, Countryname, neighbors, color=None):
    self.Countryname = Countryname
    self.neighbors = neighbors
    self.color = color
```

```
countrymap = {"A1": Continent("A1", ["A2", "H"]),
              "A2": Continent("A2", ["A3", "H", "A1"]),
              "A3": Continent("A3", ["A4", "H", "A2"]),
              "A4": Continent("A4", ["H", "A3"]),
              "H": Continent("H", ["T", "A1", "A2", "A3", "A4"]),
              "T": Continent("T", ["F1", "F2"]),
              "F1": Continent("F1", ["T"]),
              "F2": Continent("F2", ["T"])
}
```

def isCSP(x, c):

```
node=countrymap[list(countrymap.keys())[x]]
for i in node.neighbors:
    r=list(countrymap.keys()).index(i)
    if color[r] == c:
        return False
return True
```

def solve(color,v):

```
if len(countrymap)==v:
    solution= {}
    ind = 0
    for i in countrymap.keys():
        solution[i] = color[ind]
        ind += 1
    print(solution)
    return
for c in domain:
    if(isCSP(v,c)):
        color[v] = c
```

```
color[v]=0
```

```
domain = ["R","B","G"]
```

Output:

[illegible]

[illegible]