

PROIECT SINCRETIC SDA

STUDENT: CIUCLEA RAZVAN

FACULTATEA: AUTOMATICA SI CALCULATOARE

SECTIA: INFORMATICA

AN STUDIU: II

PROFESOR: DL. BOCAN VALER

TEMA PROIECTULUI:

- Un grup de N țări trebuie reprezentate pe o hartă cu culori diferite, astfel încât oricare dintre acestea să fie colorată diferit de vecinii săi. Să se scrie un program care primește la intrare lista celor N țări (denumire, listă vecini), lista de culori posibile și determină culoarea pentru fiecare țară în parte.

- În realizarea proiectului s-au folosit grafuri neorientate și metoda greedy.
- Metoda greedy face la nivel local alegerea optimă pentru fiecare etapă în speranța de a găsi un optim global.
- Altfel spus, pornim de la mulțimea vidă și adăugăm în mod repetat în B elemente până când acest lucru nu mai este posibil.
- Stabilirea elementului care va fi adăugat în soluția B se face alegându-l pe cel mai bun din acel moment — este un *optim local*.

PASII PARCURSI PENTRU REZOLVAREA PROBLEMEI

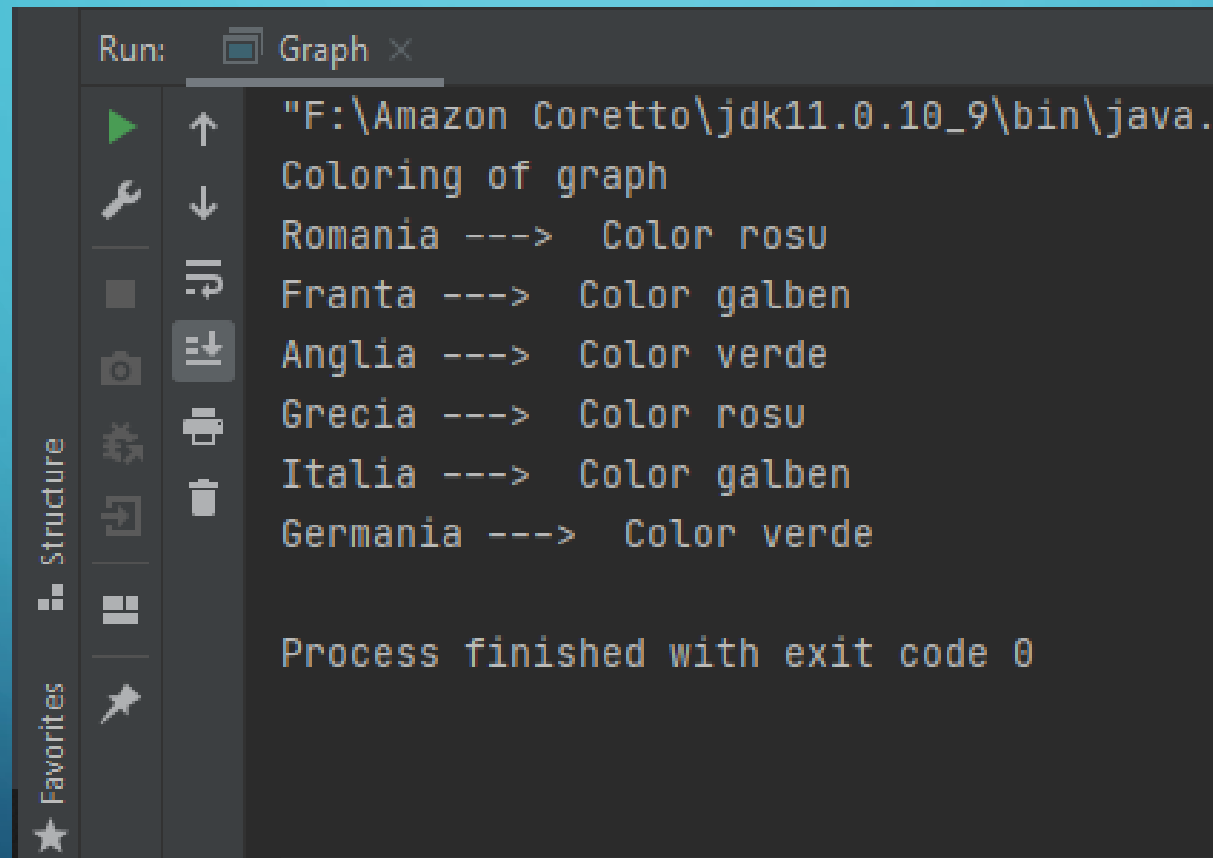
1. Se asigneaza culori, incepand din nodul 0, pentru toate nodurile.
2. Se initializeaza nodurile ca fiind neasignate si se asigneaza prima culoare primului nod.
3. Initial, toate culorile sunt disponibile si se asigneaza nodurilor ramase.
4. Se parcurg nodurile adiacente si setam flagul ca fiind indisponibil.
5. Se gaseste prima culoare disponibila, se asigneaza, iar apoi resetam valoarea inapoi la true pentru urmatoarea iteratie(false inseamna ca available[cr] este asignata unui varf adiacent)
6. Printarea rezultatului

```
Country.java x Graph.java x input.txt x
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Country {
5     private final String name;
6     private final List<Country> neighbors;
7
8     public Country(String name) {
9         this.name = name;
10        this.neighbors = new ArrayList<>();
11    }
12
13    public void addToList(Country neighbor) {
14        neighbors.add(neighbor);
15    }
16
17    public int findNeighbor(Country country) {
18        for (int i = 0; i < neighbors.size(); i++) {
19            if (neighbors.get(i).getName().equals(country.getName())) {
20                return i;
21            }
22        }
23
24        return -1;
25    }
26
27    public String getName() {
28        return name;
29    }
30
31    public List<Country> getNeighbors() {
32        return neighbors;
33    }
34 }
```

```
Country.java x Graph.java x input.txt x
1 6
2 Romania: Franta, Anglia
3 Franta: Anglia, Grecia
4 Anglia: Grecia
5 Grecia: Italia, Germania
6 Italia: Germania
7 Germania:
8 rosu, galben, verde, albastru, portocaliu, neon
```

```
Country.java x Graph.java x input.txt x
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.*;
4
5 class Graph {
6     private final int noVertices;
7     private final LinkedList<Integer>[] adj;
8     private final List<Country> countries;
9     private static List<String> colours;
10
11     public Graph(int noVertices, List<Country> countries, List<String> colours) {
12         this.noVertices = noVertices;
13         this.adj = new LinkedList<Integer>[noVertices];
14         this.countries = countries;
15         for (int i = 0; i < noVertices; ++i) {
16             this.adj[i] = new LinkedList<>();
17         }
18         Graph.colours = colours;
19     }
20
21     void addEdge(int v, int w) {
22         // graf neorientat
23         this.adj[v].add(w);
24         this.adj[w].add(v);
25     }
26
27     // asignare culori, incepand din nodul 0, pentru toate nodurile si
28     // printare culoare
29     void greedyColoring() {
30         int[] result = new int[noVertices];
31
32         // initializare noduri ca neasignate
33         Arrays.fill(result, -1);
34
35         // asignare prima culoare primului nod
36         result[0] = 0;
37
38         // valoare false inseamna ca available[i] este asignata unui varfur adiacent
39         boolean[] available = new boolean[noVertices];
```

REZULTATUL PROBLEMEI



```
Run: Graph ×
"F:\Amazon Coretto\jdk11.0.10_9\bin\java.
Coloring of graph
Romania ---> Color rosu
Franta ---> Color galben
Anglia ---> Color verde
Grecia ---> Color rosu
Italia ---> Color galben
Germania ---> Color verde

Process finished with exit code 0
```