



Problemstellung

Landkarten werden normalerweise koloriert, um einzelne Gebiete gut unterscheiden zu können. Dabei dürfen benachbarte Gebiete nicht in derselben Farbe gefärbt werden. Es sollen dabei möglichst wenige Farben verwendet werden.

1. Färbe die Karte der Bundesländer nach der beschriebenen Regel.

Viele Lösungen möglich (vgl. Simulation)



Abbildung 1: Bundesländer in Deutschland, Stefan-Xp via Wikimedia Commons (Lizenz: CC BY-SA 3.0): https://de.wikipedia.org/wiki/Datei:Germany_blank_map.svg

Modellierung

Die Ausgangssituation soll nun als Graph modelliert werden.

2. Entscheide, welche der folgenden Informationen wichtig für die Festlegung der Farben sind:
 - Name der Bundesländer
 - Wer ist wem benachbart?
 - Exakter Grenzverlauf
 - Welche Bundesländer sind Stadtstaaten?
 - Größe des Bundeslandes
 - Hat das Bundesland eine Grenze zu einem anderen Staat (z.B. Frankreich)?
 - Liegt das Bundesland an der Küste?
 - Wie viele Bundesländer gibt es?

Es ist nur wichtig, wer wem benachbart ist.

Modellierung

Knoten:

Die Knoten repräsentieren die einzelnen Regionen der Karte. Der Wert des Knotens steht für die Farbe des Knotens.

Kanten:

Zwei Knoten sind durch Kanten verbunden, wenn sie eine gemeinsame Grenze haben. Sie dürfen dann nicht in der gleichen Farbe eingefärbt werden.

Graphenfärbeproblem

Geben ist ein Graph. Färbe die Knoten des Graphen so, dass keine durch eine Kante verbundene Knoten die gleiche Farbe haben.

Varianten:

1. Verwende dabei möglichst wenige Farben.
2. Ist es möglich, den Graphen mit k Farben zu färben?





Weiterführende Fragen

Für die Kolorierung von Graphen gelten folgende Sätze:

- Graphen, die sich mit einer Farbe färben lassen, haben keine Kante außer Schleifen.
- Ein bipartiter Graph lässt sich mit zwei Farben färben.
- Ein vollständiger Graph mit n Knoten benötigt n Farben.
- Ein Graph mit einer Clique aus m Knoten benötigt mindestens m Farben.

Aufgaben:

1. Begründe diese Aussagen.

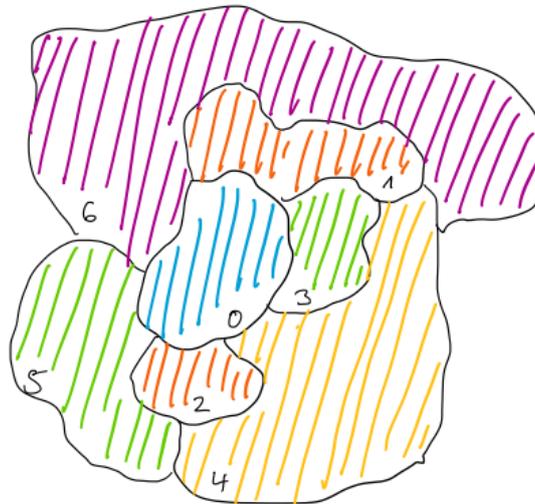
Sobald eine Kante vorhanden ist, sind zwei Knoten verbunden. Wenn dies verschiedene Knoten sind (also keine Schleife), dann dürfen diese nicht die selbe Farbe haben und es werden mindestens zwei Farben benötigt.

Wenn der Graph bipartit ist, dann zerfällt er in zwei Teilmenge der Knoten, die untereinander überhaupt nicht verbunden sind. Damit kann jede Teilmenge in einer einzigen Farbe eingefärbt werden.

Bei einem vollständigen Graphen ist jeder Knoten mit allen anderen verbunden. Daher muss jeder Knoten eine eigene Farbe haben.

In einer Clique sind alle Knoten untereinander verbunden. Daher muss jeder Knoten der Clique eine eigene Farbe bekommen.

2. Beschreibe eine Situation (Landkarte incl. Reihenfolge der Länder), in der der Greedy-Algorithmus mehr als 4 Farben erfordert. .



Werden die Knoten in der angegebenen Reihenfolge in den Graph eingefügt, findet der Greedy-Algorithmus (Farbreihenfolge: blau-rot-grün-gelb-lila) die gezeigte Lösung mit 5 Farben, da das Land 2 ungeschickterweise mit rot statt grün gefärbt wurden.

3. Landkarten lassen sich immer mit vier Farben einfärben. Bestimme die Anzahl der möglichen Färbungen (ohne Beachtung der Regel, dass Nachbarländer nicht die gleiche Farbe haben dürfen) einer Landkarte aus 20 Ländern mit 4 Farben.

Es gibt $4^{20} = \text{ca. } 1 \text{ Billion}$ Möglichkeiten.

4. Für das Kartenfärbeproblem ist kein Algorithmus bekannt, der eine optimale Lösung bestimmt, ohne dabei alle Möglichkeiten auszuprobieren. Begründe die Notwendigkeit eines Näherungsalgorithmus.

Die Anzahl der Möglichkeiten steigt mit der Anzahl der Knoten exponentiell. Damit wird der Zeitbedarf schon bei relativ wenigen Knoten zu groß. Ein Näherungsalgorithmus arbeitet mit polynomieller Zeitbedarf, liefert dafür allerdings nicht die optimale Lösung.

5. Erläutere, durch welche besondere Eigenschaft ein Graph, der eine Landkarte repräsentiert, die Beschränkung auf 4 Farben ermöglicht.

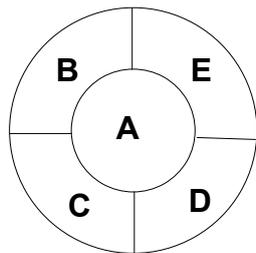


Der Graph ist planar. Durch die besondere Situation kann es keine nicht auflösbaren Kreuzungen der Kanten geben, da dies bedeuten würde, dass zwei verschiedene Ländergrenzen über Kreuz liegen.

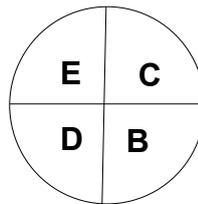
Eine Variante des Problems ist das Kolonialproblem: einige Länder haben Kolonien, die nicht direkt mit dem Mutterland verbunden sind. Dabei sollen das Mutterland und seine Kolonien in der gleichen Farbe eingefärbt werden. Weiterhin soll gelten, dass benachbarte Länder bzw. Kolonien nicht in der gleichen Farbe eingefärbt werden dürfen. Es ist nicht bekannt, ob es eine Obergrenze der Anzahl der benötigten Farben gibt.

6. Entwirf eine Landkarte mit Kolonien, bei der eine Kolorierung mit vier Farben nicht möglich ist.

z.B.



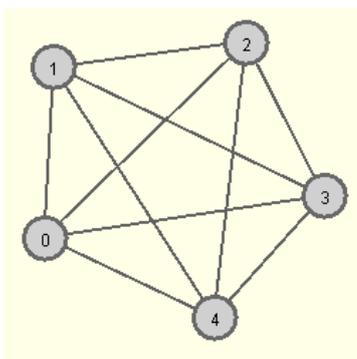
Mutterländer



Kolonien von C und E
auf einer Insel

E kann nicht die Farbe von A, B oder D haben, da dies bei den Mutterländern nicht zulässig ist. Die Farbe von C ist auch nicht zulässig, da dies bei den Kolonien nicht erlaubt ist. Keines dieser vier Länder kann die gleiche Farbe haben, da sie oder ihre Kolonien untereinander benachbart sind. Daher wird eine 5. Farbe benötigt.

7. Überführe die Karte in den dazugehörigen Graphen. Erläutere, wie Du die Forderung modellierst, dass das Mutterland und seine Kolonien in der gleichen Farbe gefärbt werden sollen.



Die Mutterländer und ihre Kolonien werden durch einen einzigen Knoten repräsentiert. Hier entsteht ein vollständiger Graph mit fünf Knoten.

8. Begründe, warum bei diesem Problem die Obergrenze von vier Farben nicht mehr gilt.

Durch die Einführung der Kolonien kann es passieren, dass der Graph nicht mehr planar ist.



Beschreibung des Algorithmus (Greedy)

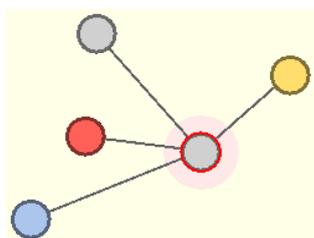
Der hier beschriebene Algorithmus findet nicht die perfekte Lösung, d.h. die minimale Anzahl an Farben, aber eine gute Näherungslösung. Er arbeitet dabei nach dem Greedy-Verfahren, er wählt für ein Land die momentan am besten erscheinende Lösung.

Zunächst wird eine Reihenfolge festgelegt, in der die Farben verwendet werden sollen.

z.B. Rot - Blau - Grün - Gelb - Lila - Orange - Braun (es müssen ausreichend viele Farben sein)

Dann betrachtet man der Reihe nach alle Knoten. Für jeden Knoten wird dann Folgendes gemacht: Man schaut jeden der Nachbarknoten an und merkt sich, dass seine Farbe schon verwendet wurde. Dann wählt man aus der Liste der Farben die erste noch nicht benutzte Farbe aus und färbt den Knoten in dieser Farbe.

z.B. Der rot umrandete Knoten ist aktuell an der Reihe. Alle Nachbarknoten werden betrachtet und ihre Farben ermittelt.



| Farbe | Rot | Blau | Grün | Gelb | Lila | Orange | Braun |
|---------|-----|------|------|------|------|--------|-------|
| Benutzt | ja | ja | nein | ja | nein | nein | nein |

Die erste noch nicht benutzte Farbe ist grün. Daher wird der Knoten grün gefärbt.



Pseudocode des Algorithmus (Greedy-Verfahren)

Der hier beschriebene Algorithmus findet nicht die perfekte Lösung, d.h. die minimale Anzahl an Farben, aber eine gute Näherungslösung. Er arbeitet dabei nach dem Greedy-Verfahren, er wählt für ein Land die momentan am besten erscheinende Lösung.

Kartenfärbung:

Wiederhole für jeden Knoten k des Graphen

 Wiederhole für jede Farbe der Farbliste
 Setze die Farbe auf "unbenutzt"
 Ende-Wiederhole

 Wiederhole für jeden Nachbarknoten n von k
 Betrachte die Farbe des Knoten n
 Setze diese Farbe auf "benutzt"
 Ende-Wiederhole

 Wiederhole für jede Farbe der Farbliste
 Falls die Farbe "unbenutzt" ist
 Färbe den Knoten k mit dieser Farbe
 Brich die Schleife ab
 Ende-Falls
 Ende-Wiederhole

Ende-Wiederhole



Quelltext des Algorithmus (Greedy-Verfahren)

```
List<Knoten> knoten = g.getAlleKnoten();
for (Knoten aktuellerKnoten: knoten) {
    boolean[] farbenliste = new boolean[g.getAnzahlKnoten()+1];
    for (int i=0; i < farbenliste.length; i++){
        farbenliste[i]=false;
    }

    List<Knoten> nachbarn = g.getNachbarknoten(aktuellerKnoten);
    for (Knoten k : nachbarn){
        farbenliste[k.getFarbe()]=true;
    }

    for (int i=1; i<farbenliste.length; i++){
        if (!farbenliste[i]) {
            aktuellerKnoten.setFarbe(i);
            break;
        }
    }
}
```