Einführung in den Graphentester

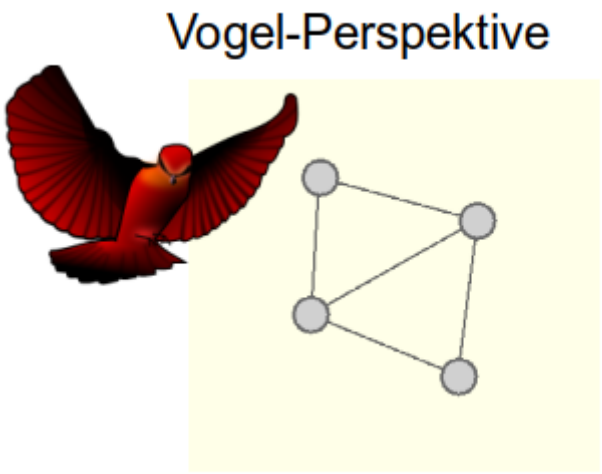
Hat ein gegebener Graph einen Eulerkreis?

Um entscheiden zu können, ob ein gegebener Graph einen Eulerkreis besitzt oder nicht, müssen wir zwei Kriterien überprüfen:

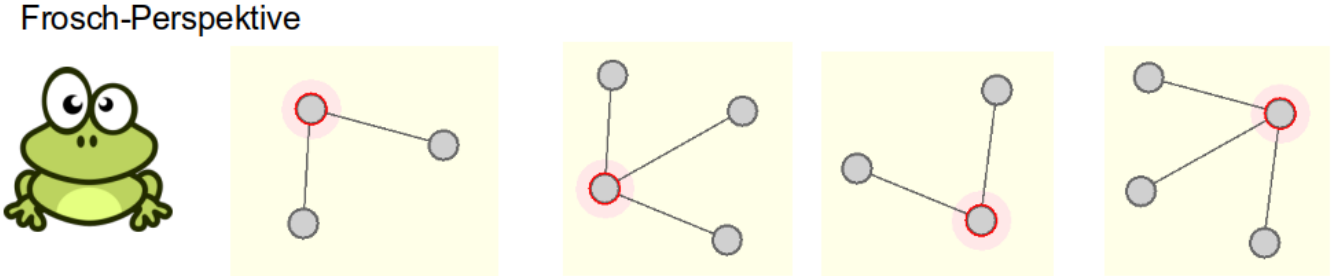
- Alle Knotengrade müssen gerade sein
- Der Graph muss zusammenhängen

Die Froschperspektive

Bei Algorithmen, die auf Graphen operieren, müssen wir einen Perspektivwechsel vornehmen: Wir sehen einen Graphen aus der "Vogelperspektive", d.h. wir nehmen den gesamten Graphen mit all seinen Knoten und Kanten wahr und lassen dann nur unsere Augen wandern - wenn uns eine Information fehlt, schauen wir einfach an die entscheidende Stelle und das Problem ist gelöst.

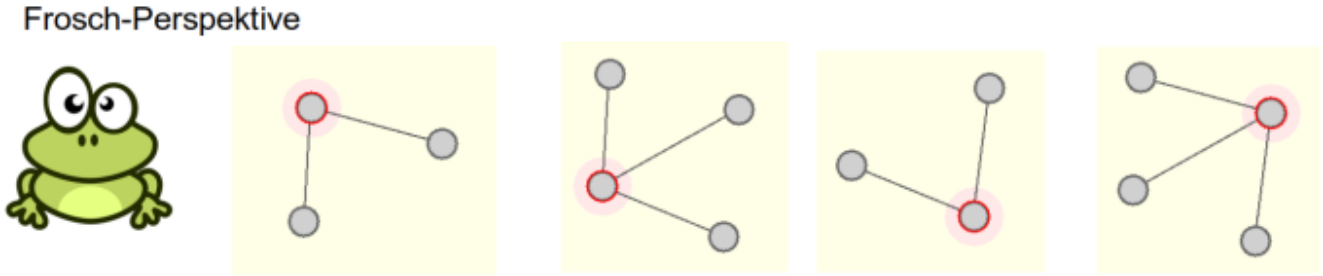


Wenn wir einen Graphen algorithmisch verarbeiten wollen, müssen wir schrittweise durch die Knotenanordnung wandern - wir sehen niemals weiter als bis zum Ende der nächsten Kante. Wir müssen also die **Frosch-Perspektive** einnehmen.



Überprüfung des Knoten-Grades

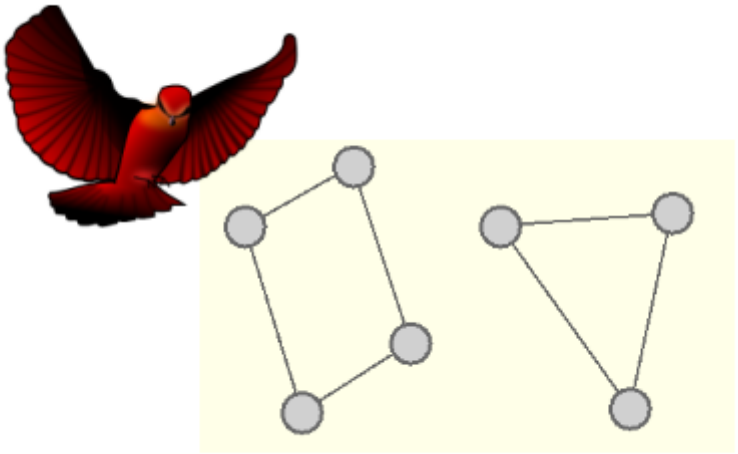
```
Setze gradOK auf true
Für jeden Knoten k:
   Wenn Grad(k) ungerade, setze gradOK auf false
```



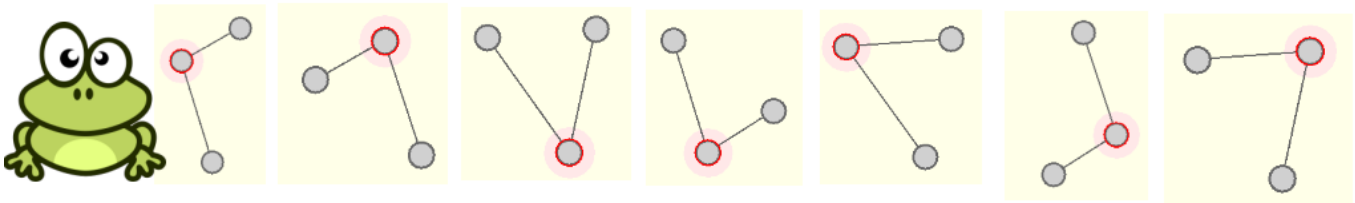
gradOK=false

Überprüfung des Zusammenhangs

Aus der Vogelperspektive ist das sofort klar - dieser Graph hängt nicht zusammen:



Der Frosch ist allerdings etwas ratlos:



(A1)

Finde eine algorithmische Vorgehensweise, wie der Frosch herausfinden kann, ob der Graph zusammenhängend ist. Um dieses Ziel zu errreichen, hat der Frosch die folgenden Werkzeuge und Fähigkeiten zur Verfügung.

Der Frosch kann...

1. Knoten / Kanten markieren
2. Bei Knoten eine Zahl eintragen
3. Knoten einfärben
4. Alle Knoten der Reihe nach durchgehen
5. Einen beliebigen Knoten als Startknoten wählen
6. Zum Nachbarknoten springen
7. Eine ToDo-Liste noch zu bearbeitender Knoten anlegen

Lade dir den Graphentester herunter und versuche einen Algorithmus zu finden.

Hilfestellung 1

Man startet bei einem beliebigen Knoten. Diesem gibt man die Nummer 1 und markiert ihn als fertig bearbeitet. Alle seine Nachbarknoten fügt man einer ToDo-Liste hinzu und kennzeichnet sie als besucht, um auszudrücken, dass sie der ToDo-Liste schon hinzugefügt, aber noch nicht fertig bearbeitet sind. Dann nimmt man den ersten Knoten aus der ToDo-Liste, nummeriert ihn und markiert ihn als fertig bearbeitet. Alle seine nicht als besucht oder fertig bearbeitet gekennzeichneten Nachbarn fügt man der ToDo-Liste hinzu und kennzeichnet sie als besucht. Diese Schritte werden so lange wiederholt, bis die ToDo-Liste leer ist. Entspricht die Nummer des zuletzt nummerierten Knotens der Anzahl der Knoten im Graph, ist der Graph zusammenhängend. Ist sie kleiner, hat man nicht alle Knoten erreicht und der Graph besteht aus mehreren Zusammenhangskomponenten. Varianten: Beim Einfügen in die ToDo-Liste können neue Knoten entweder am Anfang der Liste oder am Ende der Liste eingefügt werden.

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:graphen:zpg:eulerzug:start?rev=1668027970

Last update: **09.11.2022 21:06**