**Aufgaben zur Veranstaltung**
**Algorithmen und Datenstrukturen, SS 2020**

H. Pflug, J. Dietel FH Aachen, Campus Jülich; IT Center, RWTH Aachen

# Hausaufgaben 7 11./12.05.2020

Abgabe der Lösung am 17.05.2020

## Aufgabe 1

Verwenden Sie die Klasse `Graph` aus Präsenzübung 6 erneut als Vorlage. Ergänzen Sie nun die folgenden Methoden:

```
public boolean isDirected()

public int[] getDistances(int start)
```

- `isDirected` gibt zurück, ob der Graph gerichtet (**`true`**) oder ungerichtet (**`false`**) ist. Tipp: Sie können die Adjazenzliste von oben nach unten durchlaufen und sich damit die Breiten- oder Tiefensuche ersparen.
- `getDistances` gibt ein Feld mit den Entfernungen aller Knoten zum Start-Knoten zurück.
  - Das Feldelement 0 (Dummy-Knoten) soll den Wert -2 erhalten.
  - Alle nicht erreichbaren Knoten erhalten den Wert -1.
  - Der Startknoten selbst erhält den Wert 0.
  - Alle anderen Feldelemente erhalten die Anzahl der Kanten der kürzesten Wegstrecke vom Startknoten zum entsprechenden Knoten.
  - Tipp: Verwenden Sie die Breitensuche bei Ihrem Algorithmus.

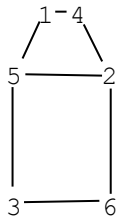
## Aufgabe 2

Zur Klasse Graph gehört auch die folgende Methode:

```
public int[] getColours() {
      int[] ret = new int[this.getVertexCount() + 1];
      for (int k = 1; k < ret.length; k++) {                    //Schleife 1
            if (ret[k] == 0) {
                  ret[k] = 1;
                  for (int i = 1; i < ret.length-1; i++) {      //Schleife 2
                        boolean changed = false;
                        for (int j = 0; j < ret.length; j++) { //Schleife 3
                              if (ret[j] > 0) {
                                    int other = 3 - ret[j];
                                    for (int next : adj[j]) {  //Schleife 4
                                          if (ret[next] == 0) {
                                                ret[next] = other;
                                                changed   = true;
                                          } else if (ret[next] != other) {
                                                return null;
                                          }
                                    }
                              }
                        }
                        if (! changed) break;
                  }
            }
      }
      return ret;
}
```

`getColours` versucht, den Graphen in zwei Farben einzufärben (1 und 2), wobei angrenzende Knoten nicht in derselben Farbe eingefärbt sein dürfen. Falls das nicht möglich ist (der Graph ist nicht bipartit), wird `null` zurückgegeben. Für die Kürze hat der Code eine beachtliche Schachtelungstiefe.

- Gegeben sei der folgende Graph:




- Im Programm wird dazu nur ein Durchlauf der 1. Schleife benötigt. Die zweite Schleife benötigt darin 3 Durchläufe. Zeichnen Sie nach jedem Durchlauf den Graphen mit der Farbe der bereits eingefärbten Knoten.
- Zeichnen Sie einen bipartiten Graphen mit 6 Elementen, bei dem die erste Schleife 1 Durchlauf und die zweite Schleife 2 Durchläufe benötigt.
- Zeichnen Sie einen bipartiten Graphen mit 6 Elementen, bei dem die erste Schleife 1 Durchlauf benötigt und die zweite Schleife die maximale Anzahl an Durchläufen.
- Zeichnen Sie einen bipartiten Graphen mit 6 Elementen, bei dem die erste Schleife 2 Durchläufe benötigt.

## Aufgabe 3

Beschreiben Sie den Lösungsalgorithmus von Aufgabe 2 aus Vorlesung 12 (Folie 312, Vorlesungsvideo ab Minute 19:40) in eigenen Worten. Hier die Aufgabenstellung:

- Studierende einer Hochschule wollen ein Benefiz-Rugby-Spiel organisieren. Aus Sorge vor Repressalien sollen Studenten nur gegen Professoren spielen, bei denen Sie keine Prüfung mehr ablegen müssen. Umgekehrt sollen, im Sinne der Teambildung, in einem Team nur Professoren sein, die allen Studenten ihres Teams noch mindestens eine Prüfung abnehmen müssen.
- Entwickeln Sie einen Algorithmus, der entscheidet, ob die Teilnehmer unter diesen Voraussetzungen grundsätzlich auf zwei Teams aufgeteilt werden können.
- Ihr Algorithmus sollte eine Komplexität von O(n + r) haben, wobei n die Zahl der Teilnehmer und r die Anzahl der Paarungen Professor, Student ist, die sich in keiner Prüfung mehr begegnen werden.