

### 3. Programmieraufgabe Computerorientierte Mathematik II

**Abgabe: 20.5.2022** über den Comajudge bis 17:00 Uhr

#### Aufgabenstellung

In dieser Aufgabe sollen Sie eine Funktion `top_order(G)` schreiben, die mittels Tiefensuche eine topologische Sortierung eines einfachen gerichteten Graphen bestimmt, falls eine existiert. Der Graph soll dabei als Liste von Objekten vom Typ `Node` gegeben sein. Hier hat jedes `Node`-Objekt als Attribute

- eine Liste `successors` aller von diesem Knoten aus *direkt* erreichbaren Knoten,
- einen String `name` und
- einen String `color`, der bei Initialisierung auf `"white"` gesetzt wird.

Es darf davon ausgegangen werden, dass alle Knoten unterschiedliche Namen haben.

**Eingabe** Es wird eine Liste  $G$  von `Node`-Objekten übergeben, die einen gerichteten Graphen  $G$  repräsentiert.

**Ausgabe** Besitzt  $G$  eine topologische Sortierung, so werden die `name`-Werte der Knoten topologisch sortiert als Liste  $[v_1, \dots, v_n]$  zurückgegeben. Andernfalls wird die Liste `[-1]` zurückgegeben.

*Hinweis:* Der Graph muss nicht zusammenhängend sein. Ist  $n = 0$ , so wird eine leere Liste zurückgegeben. Verwenden Sie zum Testen Ihrer Funktion eine Klasse `Node` mit den oben angegebenen Attributen. Außerdem bietet es sich zum Ausprobieren mehrerer Beispiele an, eine Funktion zu schreiben, die anhand einer einfachen Datenstruktur für Graphen (Adjazenz-/Inzidenzliste/-matrix) eine Liste von Knotenobjekten mit den korrekten `successors`-Listen erstellt.

#### Beispielaufrufe

```

1>>> n = Node()
2>>> m = Node()
3>>> n.name = "Quelle"
4>>> m.name = "Senke"
5>>> n.color = m.color = "white"
6>>> n.successors = [m]
7>>> m.successors = []
8>>> G = [m,n]
9>>> top_order(G)
10 ['Quelle', 'Senke']
11>>> n = Node()
12>>> m = Node()
13>>> n.name = "links"
14>>> m.name = "rechts"
15>>> n.color = m.color = "white"
16>>> n.successors = [m]
17>>> m.successors = [n]
18>>> G = [m,n]
19>>> top_order(G)
20 [-1]
```