

B.14:

Sei G ein Digraph.

1. Schreiben Sie eine PYTHON-Funktion `reverse(G)`, die den Umkehrgraph G^{rev} von G zurückliefert. Der Graph G darf dabei nicht verändert werden.

```
1  # verwendet dfs
2  from operator import truediv
3  from vl.lek03.dfs import dfs
4  from vl.lek03.bfs import bfs
5
6  # Umkehrgraph eines Digraphen
7  def reverse(G):
8      m = len(G)
9      Grev = [set() for _ in range(m)] # alle Mengen leer anlegen
10
11     for i in range(m):                # Eintrag und Index vertauschen
12         for v in G[i]:
13             Grev[v].add(i)
14     return Grev
```

2. Implementieren Sie eine PYTHON-Funktion `strong_comp(G)`, die eine Liste aller *starken* Zusammenhangskomponenten von G zurückliefert. Die Ergebnisliste enthält die starken Zusammenhangskomponenten als Knotenmengen, aufsteigend sortiert nach minimaler Knotennummer der Menge.

```
16  # alle starken Zusammenhangskomponenten eines Digraphen
17  def strong_comp(G):
18      C = []
19
20      Grev = reverse(G) # Umkehrgraph von G
21      m = len(G)
22      reached = [False]*m # bereits erreicht Knoten
23      node_available = True # zeigt an ob noch unerreichte Knoten im Graph existieren
24
25      while node_available:
26          v = getUnreachedNode(reached)
27          if(v != -1): # Es existieren noch unerreichte Knoten im Graphen
28              result = dfs(G,v) # Tiefensuche von v in Graph und Umkehrgraph
29              result_rev = dfs(Grev,v)
30
31              intersection = result[0] & result_rev[0] # Schnittmenge der erreichten Knoten von Tiefensuche
32
33              # Alle erreichten Knoten in Zusammenhangskomponente speichern
34              # und erreichten Knoten in reached markieren
35              for u in intersection:
36                  reached[u] = True
37
38              C.append(intersection)
39          else: # Alle Knoten des Graphen wurden besucht
40              node_available = False
41
42      return C
```

```
44  def getUnreachedNode(reached):
45      for i in range(len(reached)):
46          if reached[i] == False:
47              return i
48
49      return -1
50
```