Übungsblatt 5.

Bearbeiten bis: **Montag, 07.04.2025**

Aufgabe 1 (Topologische Ordnung | 4 Punkte). Eine topologische Ordnung eines Digraphen $G = (V, E)$ ist eine Ordnung der Knoten $V = \{v_1, \ldots, v_n\}$, so dass für jede Kante $e = (v_i, v_j) \in E$ die Relation $i < j$ gilt.

Zeigen Sie, dass ein Digraph genau dann eine topologische Ordnung hat, wenn er azyklisch ist.

Aufgabe 2 (Anzahl Kanten). Sei $G = (V, E)$ ein Graph mit $n = |V| \geq 3$ Knoten. Ausserdem sei $G'(v) = (V'(v), E'(v))$ derjenige Graph, der aus G durch Streichung von $v \in V$ sowie aller an v anliegenden Kanten entsteht. Zeigen Sie, dass dann folgende Aussagen gelten:

(a) $|E| = \frac{1}{n-2} \sum_{v \in V} |E'(v)|$.

(b) Falls $n > 3$ und $|E| > \frac{n^2}{4}$ gilt, so gibt es einen Knoten $v \in V$, so dass $G'(v)$ mehr als $\frac{(n-1)^2}{4}$ Kanten hat.

Aufgabe 3 (Starker Zusammenhang | 4 Punkte). Wenden Sie Algorithmus 3.7 aus der Vorlesung auf folgenden Digraphen G an, um die starken Zusammenhangskomponenten zu bestimmen. Stellen Sie dabei den Digraphen G mit einer Nachbarschaftsmatrix dar. In der Nachbarschaftsmatrix sollen zu einem Knoten $v \in V$ die Nachbarknoten in aufsteigender Reihenfolge abgelegt werden. Erläutern Sie für jeden Schritt, welche Knoten in R sind sowie die Werte der Funktionen ι und comp. Was für eine Kardinalität haben die erhaltenen Zusammenhangskomponenten? Zeichnen Sie schliesslich den kondensierten Digraphen.

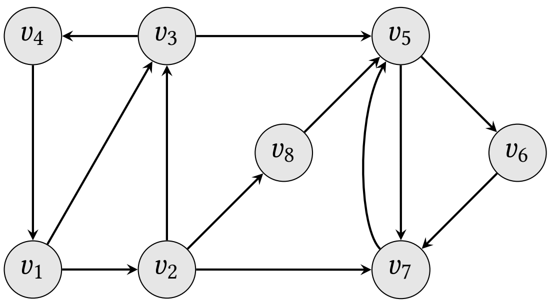

Abbildung 1: Digraph G.

Aufgabe 4 (Implementierung starker Zusammenhang). Implementieren Sie den folgenden Algorithmus zur Bestimmung der starken Zusammenhangskomponenten eines Digraphen, und wenden Sie ihn dann auf den Digraphen G in Abbildung 1 an.

Hinweis. ι und ι^{-1} können Sie als Vektoren implementieren. Wenn Sie die Hilfsfunktionen (visit1, visit2, Ainv) in MATLAB als *nested functions* implementieren, haben diese Zugriff auf die Variablen (R, N, C etc.) in der Elterfunktion (strongComp).

Algorithmus strongComp (**A**)

Input: Nachbarschaftsmatrix **A** von Digraph G
Output: Vektor c der Länge $|V|$, der für jeden Knoten die Zugehörigkeit zur starken Zusammenhangskomponente kennzeichnet

```
R = 0
N = 0
for all v ∈ V do
   if v ∉ R then
      visit1(v)
   end if
end for

R = 0
K = 0
Ā = Ainv()
for j = (|V|, |V| − 1, ..., 1) do
   if ι^{-1}(j) ∉ R then
      K = K + 1
      visit2(ι^{-1}(j))
   end if
end for

% Hilfsfunktion 1
function visit1(v)
   R = [R; v]
   for all w ∈ A[ : ][v] do
      if w ∉ R then
         visit1(w)
      end if
   end for
   N = N + 1
   ι(v) = N
   ι^{-1}(N) = v
end function

% Hilfsfunktion 2
function visit2(v)
   R = [R; v]
   for all q ∈ Ā[ : ][v] do
      if q ∉ R then
         visit2(q)
      end if
   end for
   c(v) = K
end function

% Hilfsfunktion für Nachbarschaftsmatrix Ā des inversen Digraph G^{-1}
function Ā = Ainv()
   for all k = 1 : |V| do
      for all l ∈ A[ : ][k] mit l ≠ 0 do
         Ā = addEdge(Ā, l, k)
      end for
   end for
end function
```