Graphen-Grundbegriffe

Ein endlicher Graph $G = (V, E)$ besteht aus
einer endlichen Menge von Knoten V (vertices)
und
einer endlichen Menge von Kanten E (edges),
$E \subseteq V \times V$.

Sei G ein Graph mit Knotenmenge
$V = \{v_0, \ldots, v_{n-1}\}$.
Die Einträge a_{ij} (0$\leqslant i,j \leqslant$n-1) der

$n \times n$-Adjazenzmatrix zu G

sind definiert durch

$$a_{ij} := \begin{cases} 1 & \text{falls } (v_i, v_j) \in E \\ 0 & \text{sonst} \end{cases}$$

Ein Graph G heißt ungerichtet, falls zu jedem $(v, v') \in E$ auch $(v', v) \in E$, ansonsten gerichtet.
Die Adjazenzmatrix eines ungerichteten Graphen ist symmetrisch.

Gewichtete Graphen

Ein Graph G heißt gewichtet, falls jeder Kante mittels einer Gewichtsfunktion

$$w : E \to \mathbb{R}_0^+$$

eine nicht-negative reelle Zahl zugeordnet ist.

w kann zu $\tilde{w} : V \times V \to \mathbb{R}_0^+ \cup \{\infty\}$ ergänzt werden:

$$\tilde{w}(v_i, v_j) = \begin{cases} w(v_i, v_j) & \text{falls } (v_i, v_j) \in E \\ \infty & \text{sonst} \end{cases}$$

Je nach Anwendung kann statt ∞ auch 0 gewählt werden .

Pfade, Zykel und Wege

Eine Folge von Kanten

$$(v_{i_1}, v_{i_2})(v_{i_2}, v_{i_3}) \dots (v_{i_k}, v_{i_{k+1}})$$

heißt Pfad, falls alle Knoten $v_{i_1} \dots v_{i_{k+1}}$ der Folge voneinander verschieden sind.

Eine Kantenfolge $(v_{i_1}, v_{i_2})(v_{i_2}, v_{i_3}) \dots (v_{i_k}, v_{i_1})$ heißt Zykel, falls alle Knoten $v_{i_1} \dots v_{i_k}$ voneinander verschieden sind.

Eine Kantenfolge $(v_{i_1}, v_{i_2})(v_{i_2}, v_{i_3}) \dots (v_{i_k}, v_{i_{k+1}})$ heißt Weg, falls alle Kanten voneinander verschieden sind.

Ein Graph ohne Zykel heißt azyklisch.

Subgraphen, Zusammenhang und Bäume

Ein Graph (V', E') heißt Subgraph von $G = (V, E)$, falls $V' \subseteq V$ und $E' \subseteq E$.

Ein ungerichteter Graph heißt zusammenhängend, falls zu jedem Knotenpaar v_i und v_j in G ein Pfad von v_i nach v_j existiert.

Ein Baum ist ein zusammenhängender, ungerichteter, azyklischer Graph.

Ein aufspannender Baum eines Graphen G ist ein Subgraph, der alle Knoten von G umfasst und ein Baum ist.

Zusammenhangskomponenten

Zusammenhangskomponenten eines Graphen sind zusammenhängende Subgraphen mit maximaler Knotenzahl.

Sei $G = (V, E)$ mit $V = \{v_0, \ldots, v_{n-1}\}$ ungerichtet.
Die $n \times n$-Zusammenhangsmatrix

$$C = (c_{ij})_{0 \leqslant i,j \leqslant n-1}$$

wird definiert durch

$$c_{ij} := \begin{cases} 1 & \text{falls } v_i \text{ und } v_j \text{ durch einen Weg} \\ & \text{der Länge } \geqslant 0 \text{ verbunden sind} \\ 0 & \text{sonst} \end{cases}$$

C ergibt sich als reflexiver, transitiver Abschluss der Adjazenzmatrix unter der Booleschen Matrixmultiplikation, bei der als Multiplikation die Konjunktion und als Addition die Disjunktion verwendet wird.
Statt

$$c_{ij} = \sum_{k=0}^{n-1} (a_{ik} * b_{kj})$$

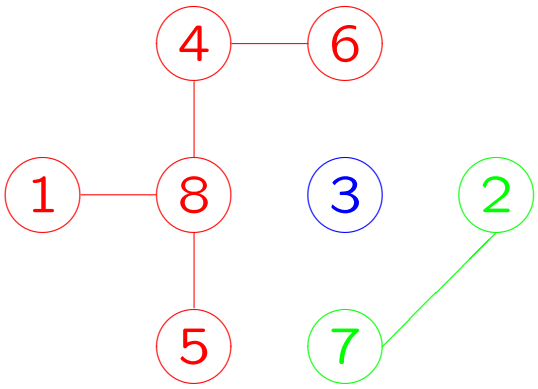
wird

$$c_{ij} = \bigvee_{k=0}^{n-1} (a_{ik} \wedge b_{kj})$$

berechnet.

Bestimmung der Zusammenhangskomponenten eines ungerichteten Graphen

Beispiel:



Adjazenzmatrix

	1	2	3	4	5	6	7	8
1								1
2							1	
3								
4						1		1
5								1
6				1				
7		1						
8	1			1	1			

Zusammenhangsmatrix $\mathcal{C} = (c_{ij})_{0 \leqslant i,j \leqslant n-1}$

	1	2	3	4	5	6	7	8
1	1			1	1	1		1
2		1					1	
3			1					
4	1			1	1	1		1
5	1			1	1	1		1
6	1			1	1	1		1
7		1					1	
8	1			1	1	1		1

Zur Zusammenhangskomponente eines Knoten v_k gehören alle Knoten v_j mit $c_{kj} = 1 = c_{jk}$.

Vektordarstellung:

	1	2	3	4	5	6	7	8
$\mathcal{C}$	1	2	3	1	1	1	2	1

Pseudocode Hirschberg-Algorithmus

Iteriere $\lceil \log n \rceil$-mal die folgenden 3 Phasen:
Initialisierung des C-Vektors:
$C(i) := i$ für alle $1 \leqslant i \leqslant n$.

Phase 1:

for all vertices i in parallel do

$$T(i) = \begin{cases} \min_j\{\, C(j) \mid & \text{falls existent} \\ \qquad A(i,j) = 1, & \\ \qquad C(j) \neq C(i)\} & \\ C(i) & \text{sonst} \end{cases}$$

Phase 2:

for all vertices i in parallel do

$$T(i) = \begin{cases} \min_j\{\, T(j) \mid & \text{falls existent} \\ \qquad C(j) = i, & \\ \qquad T(j) \neq i\} & \\ C(i) & \text{sonst} \end{cases}$$

Phase 3:

for all vertices i in parallel do $B(i) = T(i)$
repeat $\log n$ times
    for all vertices i in parallel do
        $T(i) = T(T(i))$
for all vertices i in parallel do
    $C(i) = \min\{B(T(i)), T(i)\}$

Detaillierung und Analyse von Phase 1/2

Phase 1 und 2 sind ähnlich,
hier Detaillierung von Phase 1:

1(a) for all i, j $(1 \leqslant i, j \leqslant n)$ in parallel do

$$Temp(i,j) := \text{if } A(i,j) = 1 \text{ and } C(i) \neq C(j) \text{ then } C(j) \text{ else } \infty$$

1(b) for all i $(1 \leqslant i \leqslant n)$ in parallel do

$$Temp(i,1) := \min_{1 \leqslant j \leqslant n} \{Temp(i,j)\}$$

1(c) for all i $(1 \leqslant i \leqslant n)$ in parallel do

$$T(i) := \text{if } Temp(i,1) = \infty \text{ then } C(i) \text{ else } Temp(i,1)$$

∞ repräsentiert eine Zahl $> n$.

Aufwandsanalyse von Phase 1 und 2:

1/2(a)	$O(1)$	mit $O(n^2)$ Proz.
1/2(b)	$O(\log n)$	mit $O(n^2)$ Proz.
1/2(c)	$O(1)$	mit $O(n)$ Proz.

$\curvearrowright$ $O(\log n)$ mit $O(n^2)$ Proz.

Begründung für Phase 3

Der T-Graph, der in Phase 2 gebildet wird, besitzt in jedem neu zu bildenden Superknoten eine Schleife aus 2 Kanten, wobei der kleinste Knoten des neuen Superknotens einer der beiden Schleifenknoten ist.

Da jeder Superknoten aus höchstens n Knoten besteht, kommt man beim Durchlaufen des T-Graphen nach n Schritten in die Schleife und ist somit höchstens einen Schritt vom Minimum entfernt.

Gesamtaufwand:

Phase 1/2: $O(\log n)$ mit $O(n^2)$ Proz.

Phase 3:	3(a)	$O(1)$	mit $O(n)$ Proz.
	3(b)	$O(\log n)$	mit $O(n)$ Proz.
	3(c)	$O(1)$	mit $O(n)$ Proz.

$\curvearrowright O(\log n)$ mit $O(n)$ Proz.

$\log n$ Iterationen: $\boxed{O(\log^2 n) \text{ mit } O(n^2) \text{ Proz.}}$