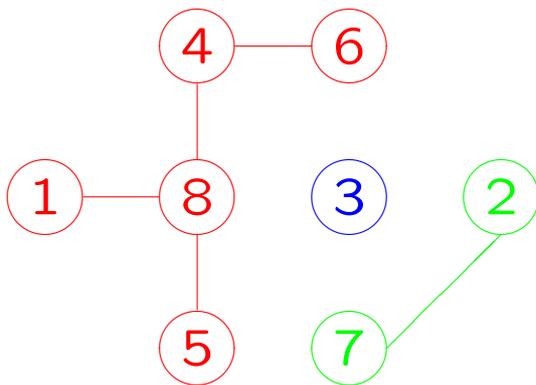


Bestimmung der Zusammenhangskomponenten eines ungerichteten Graphen

Beispiel:



Adjazenzmatrix

$$\begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & & & & & & & & 1 \\ 2 & & & & & & & 1 & \\ 3 & & & & & & & & \\ 4 & & & & & & 1 & & 1 \\ 5 & & & & & & & & 1 \\ 6 & & & & 1 & & & & \\ 7 & & 1 & & & & & & \\ 8 & 1 & & & 1 & 1 & & & \end{pmatrix}$$

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

$$\begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 1 & & & 1 & 1 & 1 & & 1 \\ 2 & & 1 & & & & & 1 & \\ 3 & & & 1 & & & & & \\ 4 & 1 & & & 1 & 1 & 1 & & 1 \\ 5 & 1 & & & 1 & 1 & 1 & & 1 \\ 6 & 1 & & & 1 & 1 & 1 & & 1 \\ 7 & & 1 & & & & & 1 & \\ 8 & 1 & & & 1 & 1 & 1 & & 1 \end{pmatrix}$$

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

Vektordarstellung: $C \begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 1 & & & 1 & 1 & 1 & & 1 \\ 2 & & 1 & & & & & 1 & \\ 3 & & & 1 & & & & & \\ 4 & 1 & & & 1 & 1 & 1 & & 1 \\ 5 & 1 & & & 1 & 1 & 1 & & 1 \\ 6 & 1 & & & 1 & 1 & 1 & & 1 \\ 7 & & 1 & & & & & 1 & \\ 8 & 1 & & & 1 & 1 & 1 & & 1 \end{array}$

Pseudocode Hirschberg-Algorithmus

Iteriere $\lceil \log n \rceil$ -mal die folgenden 3 Phasen:

Initialisierung des C -Vektors:

$C(i) := i$ für alle $1 \leq i \leq n$.

Phase 1:

for all vertices i in parallel do

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

Phase 2:

for all vertices i in parallel do

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

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 ($1 \leq i, j \leq n$) in parallel do

$Temp(i, j) :=$ if $A(i, j) = 1$ and $C(i) \neq C(j)$
then $C(j)$ else ∞

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

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

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

$T(i) :=$ if $Temp(i, 1) = \infty$ then $C(i)$
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 $O(n)$ 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.

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

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