WS 2003/04

Diskrete Strukturen I

Ernst W. Mayr

mayr@in.tum.de
Institut für Informatik
Technische Universität München

2004-01-20

Adjazenzmatrix

Definition

Sei $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$. Dann heißt

$$A = (a_{ij})_{1 \leq i,j \leq n} \quad \text{mit } a_{ij} = \begin{cases} 1 & \text{falls } \{v_i, v_j\} \in E \\ 0 & \text{sonst} \end{cases}$$

die *Adjazenzmatrix von* G.

Beobachtungen:

- Für ungerichtete Graphen ist die Adjazenzmatrix symmetrisch.
- Gibt es keine Schlingen, so sind alle Diagonalelemente null.

Adjazenzmatrix

Definition

Sei $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$. Dann heißt

$$A = (a_{ij})_{1 \leq i,j \leq n} \quad \text{mit } a_{ij} = \begin{cases} 1 & \text{falls } \{v_i, v_j\} \in E \\ 0 & \text{sonst} \end{cases}$$

die *Adjazenzmatrix von* G.

Beobachtungen:

- Für ungerichtete Graphen ist die Adjazenzmatrix symmetrisch.
- Gibt es keine Schlingen, so sind alle Diagonalelemente null.

Adjazenzmatrix

Definition

Sei $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$. Dann heißt

$$A = (a_{ij})_{1 \leq i,j \leq n} \quad \text{mit } a_{ij} = \begin{cases} 1 & \text{falls } \{v_i, v_j\} \in E \\ 0 & \text{sonst} \end{cases}$$

die *Adjazenzmatrix von G.*

Beobachtungen:

- Für ungerichtete Graphen ist die Adjazenzmatrix symmetrisch.
- Gibt es keine Schlingen, so sind alle Diagonalelemente null.

Satz

Sei A die Adjazenzmatrix von $G = (V, E)$, $|V| = n$, und sei

$$A^0 := I,$$
$$A^{i+1} := A^i \cdot A \qquad \text{für alle } i \geq 0.$$

Dann gilt für

$$A^k = \left(a_{ij}{}^{(k)}\right)_{1 \leq i,j \leq n} :$$

$a_{i,j}{}^{(k)}$ *ist die Anzahl verschiedener Pfade in G von v_i nach v_j der Länge k.*

Beweis.

Induktion nach k:

Induktionsanfang: $k = 0$ und $k = 1$ sind trivial. Induktionsschluss:

$k \mapsto k + 1$

${a_{il}}^{(k)}$ ist nach Induktionsvoraussetzung die Anzahl verschiedener Pfade der Länge k von v_i nach v_l.

Die Anzahl verschiedener Pfade von v_i nach v_j der Länge $k + 1$ lässt sich wie folgt berechnen:

$$\sum_{l=1}^{n} {a_{il}}^{(k)} \cdot a_{lj} = {a_{ij}}^{(k+1)}$$

□

Beweis.

Induktion nach k:

Induktionsanfang: $k = 0$ und $k = 1$ sind trivial. Induktionsschluss:

$k \mapsto k + 1$

${a_{il}}^{(k)}$ ist nach Induktionsvoraussetzung die Anzahl verschiedener Pfade der Länge k von v_i nach v_l.

Die Anzahl verschiedener Pfade von v_i nach v_j der Länge $k + 1$ lässt sich wie folgt berechnen:

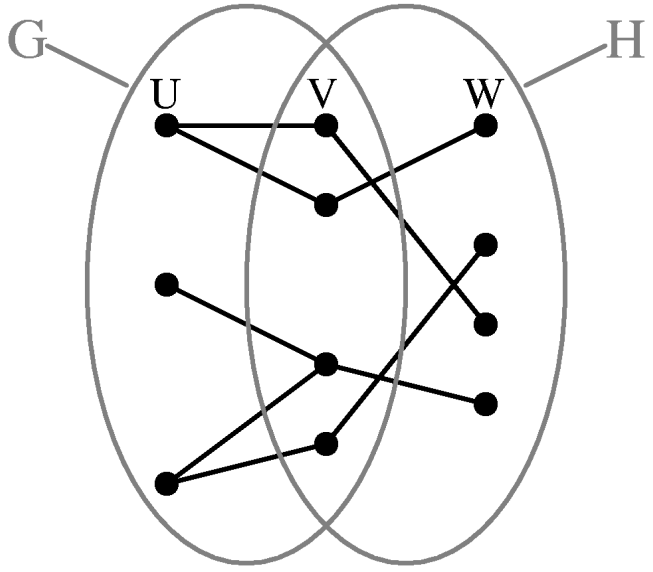
$$\sum_{l=1}^{n} {a_{il}}^{(k)} \cdot a_{lj} = {a_{ij}}^{(k+1)}$$

Bemerkung: Adjazenzmatrix von bipartiten Graphen
Sei $G = (U, V, E)$ mit $U = \{u_1, \ldots, u_n\}$ und $V = \{v_1, \ldots, v_m\}$ ein bipartiter Graph. Dann heißt

$$A = \left(a_{ij}\right)_{\substack{1 \le i \le n \\ 1 \le j \le m}} \quad \text{mit } a_{ij} = \begin{cases} 1 & \text{falls } \{u_i, v_j\} \in E \\ 0 & \text{sonst} \end{cases}$$

die *Adjazenzmatrix von* G.

Werden zwei bipartite Graphen zusammengesetzt, zum Beispiel:



berechnet sich die Adjazenzmatrix A' des bipartiten Graphen $G' = (U, W, E')$, mit

$$\{u, w\} \in E' \iff (\exists v \in V)[\{u, v\} \text{ in } G \text{ und } \{v, w\} \text{ in } H]$$

als das boolesche Produkt $A_G \cdot A_H$.

Inzidenzmatrix

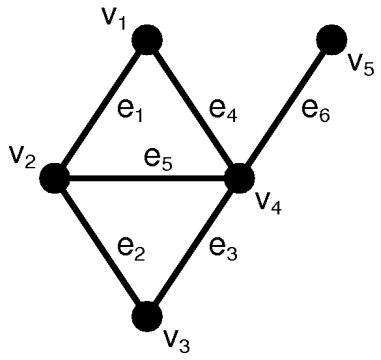
Definition

Sei $G = (V, E)$ mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_m\}$.
Dann heißt

$$B = \left(b_{ij}\right)_{\substack{1 \le i \le n \\ 1 \le j \le m}} \quad \text{mit} \quad b_{i,j} = \begin{cases} 1 & \text{falls } v_i \in e_j \\ 0 & \text{sonst} \end{cases}$$

die *Inzidenzmatrix von* G.

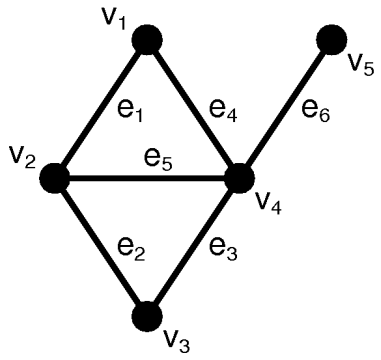
Beispiel (Adjazenz- und Inzidenzmatrix)



Adjazenzmatrix:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

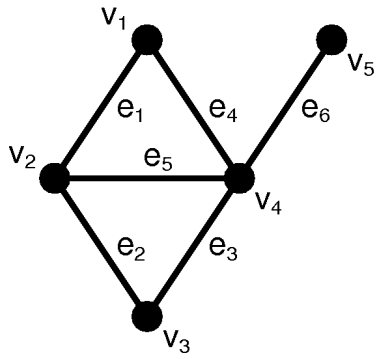
Beispiel (Adjazenz- und Inzidenzmatrix)



Adjazenzmatrix:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

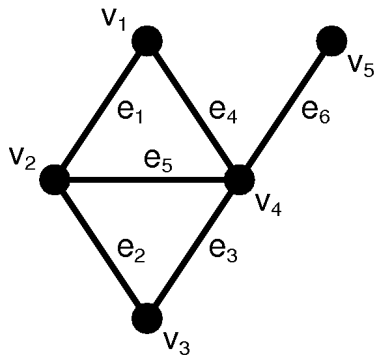
Beispiel (Adjazenz- und Inzidenzmatrix)



Adjazenzmatrix:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

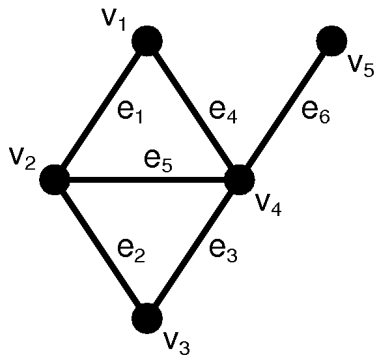
Beispiel (Adjazenz- und Inzidenzmatrix)



Inzidenzmatrix:

$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

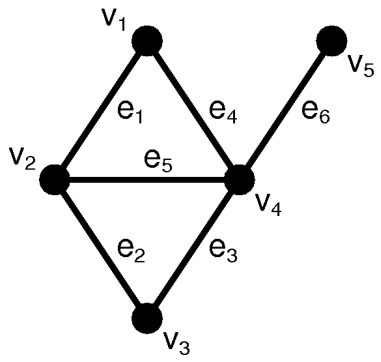
Beispiel (Adjazenz- und Inzidenzmatrix)



Inzidenzmatrix:

$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Beispiel (Adjazenz- und Inzidenzmatrix)



Inzidenzmatrix:

$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Beobachtung:

$$B \cdot B^T = \begin{pmatrix} \mathsf{d}(v_1) & & & \\ & \mathsf{d}(v_2) & & 0 \\ & & \ddots & \\ 0 & & & \mathsf{d}(v_n) \end{pmatrix} + A$$

Definitionen für gerichtete Graphen

Digraph

Definition

Ein *Digraph (aka gerichteter Graph, engl. directed graph)* $G = (V, A)$ *besteht aus einer Knotenmenge* V *und einer Menge* $A \subseteq V \times V$ *von geordneten Paaren, den gerichteten Kanten.*

Grad

Definition

- $d^-(v)$ ist der *Aus-Grad von v, d. h. die Anzahl der Kanten mit Anfangsknoten v.*
- *$d^+(v)$ ist der In-Grad von v, d. h. die Anzahl der Kanten mit Endknoten v.*
- *$d(v) = d^-(v) + d^+(v)$ ist der (Gesamt-)Grad von v.*

Grad

Definition

- $d^-(v)$ ist der *Aus-Grad von v, d. h. die Anzahl der Kanten mit Anfangsknoten v.*
- $d^+(v)$ *ist der In-Grad von v, d. h. die Anzahl der Kanten mit Endknoten v.*
- $d(v) = d^-(v) + d^+(v)$ *ist der (Gesamt-)Grad von v.*

Grad

Definition

- $\mathrm{d}^-(v)$ ist der *Aus-Grad von v, d. h. die Anzahl der Kanten mit Anfangsknoten v.*
- $d^+(v)$ *ist der In-Grad von v, d. h. die Anzahl der Kanten mit Endknoten v.*
- $d(v) = d^-(v) + d^+(v)$ *ist der (Gesamt-)Grad von v.*

Beobachtung:

$$\sum_{v \in V} \mathsf{d}^-(v) = \sum_{v \in V} \mathsf{d}^+(v) = |A|$$

Adjazenzmatrix

Definition

Sei $G = (V, A)$ ein Digraph mit $V = \{v_1, \ldots, v_n\}$. Dann heißt

$$C = \left(c_{ij}\right)_{1 \leq i,j \leq n} \quad \text{mit } c_{ij} = \begin{cases} 1 & \text{falls } (v_i, v_j) \in A \\ 0 & \text{sonst} \end{cases}$$

die *Adjazenzmatrix von* G.

Inzidenzmatrix

Definition

Sei $G = (V, A)$ ein (einfacher!) Digraph mit $V = \{v_1, \ldots, v_n\}$ und $A = \{e_1, \ldots, e_m\}$. Dann heißt

$$B = \left(b_{ij}\right)_{\substack{1 \le i \le n \\ 1 \le j \le m}} \text{ mit } b_{ij} = \begin{cases} 1 & \text{falls } v_i \text{ Endknoten von } e_{ij} \\ -1 & \text{falls } v_i \text{ Anfangsknoten von } e_{ij} \\ 0 & \text{sonst} \end{cases}$$

die *Inzidenzmatrix von* G.

Beobachtung:

$$B \cdot B^T = \begin{pmatrix} \mathsf{d}(v_1) & & & \\ & \mathsf{d}(v_2) & & 0 \\ & & \ddots & \\ 0 & & & \mathsf{d}(v_n) \end{pmatrix} - C$$

Diese Matrix heißt *Laplacesche Matrix.*

Beobachtung:

$$B \cdot B^T = \begin{pmatrix} \mathsf{d}(v_1) & & & \\ & \mathsf{d}(v_2) & & 0 \\ & & \ddots & \\ 0 & & & \mathsf{d}(v_n) \end{pmatrix} - C$$

Diese Matrix heißt *Laplacesche Matrix.*

Gerichteter Pfad

Definition

Eine Folge $(u_0, u_1, \ldots, u_n)$ mit $u_i \in V$ für $i = 0, \ldots, n$ heißt *gerichteter Pfad, wenn*

$$\big(\forall i \in \{0, \ldots, n-1\}\big)\Big[(u_i, u_{i+1}) \in A\Big].$$

Ein gerichteter Pfad heißt einfach, falls alle u_i paarweise verschieden sind.

Gerichteter Pfad

Definition

Eine Folge $(u_0, u_1, \ldots, u_n)$ mit $u_i \in V$ für $i = 0, \ldots, n$ heißt *gerichteter Pfad, wenn*

$$\big(\forall i \in \{0, \ldots, n-1\}\big)\Big[(u_i, u_{i+1}) \in A\Big].$$

Ein gerichteter Pfad heißt einfach, falls alle u_i paarweise verschieden sind.

Gerichteter Kreis

Definition

Ein gerichteter Pfad $(u_0, u_1, \ldots, u_n)$ heißt *gerichteter Kreis, wenn* $u_0 = u_n$.

Der gerichtete Kreis heißt einfach, falls $u_0, u_1, \ldots, u_{n-1}$ *alle paarweise verschieden sind.*

dag s

Definition

Ein Digraph, der keinen gerichteten Kreis enthält, heißt *directed acyclic graph, kurz dag.* *In einem dag heißen Knoten mit In-Grad 0 Quellen, Knoten mit Aus-Grad 0 Senken. Eine Nummerierung $i : V \to \{1, \ldots, |V|\}$ der Knoten eines dags heißt topologisch, falls für jede Kante $(u, v) \in A$ gilt:*

$$i(u) < i(v).$$

dag s

Definition

Ein Digraph, der keinen gerichteten Kreis enthält, heißt *directed acyclic graph, kurz dag. In einem dag heißen Knoten mit In-Grad 0 Quellen, Knoten mit Aus-Grad 0 Senken. Eine Nummerierung* $i : V \to \{1, \ldots, |V|\}$ *der Knoten eines dags heißt topologisch, falls für jede Kante* $(u, v) \in A$ *gilt:*

$$i(u) < i(v).$$

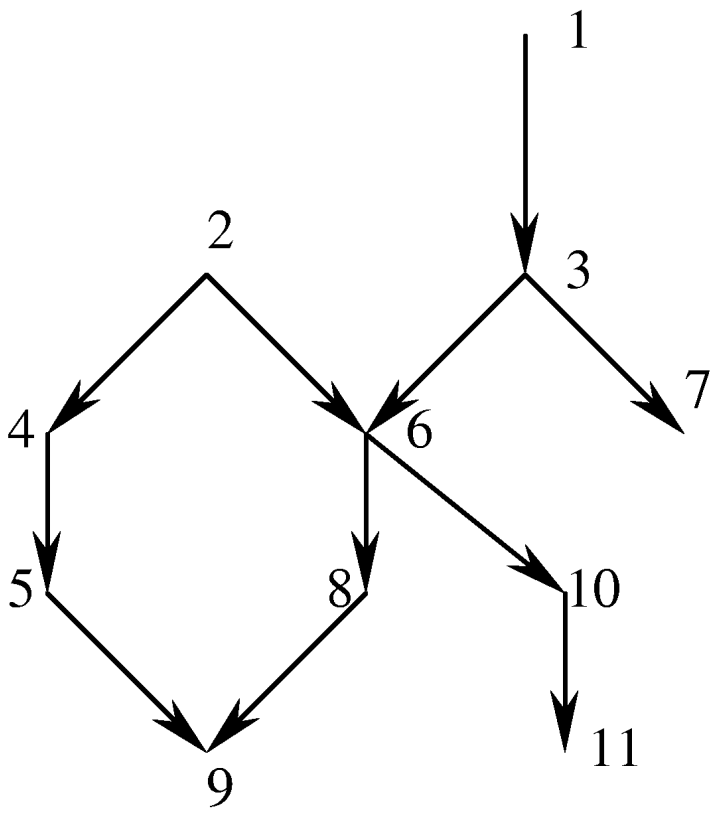
dag s

Definition

Ein Digraph, der keinen gerichteten Kreis enthält, heißt *directed acyclic graph, kurz dag. In einem dag heißen Knoten mit In-Grad 0 Quellen, Knoten mit Aus-Grad 0 Senken. Eine Nummerierung* $i : V \to \{1, \ldots, |V|\}$ *der Knoten eines dags heißt topologisch, falls für jede Kante* $(u, v) \in A$ *gilt:*

$$i(u) < i(v).$$

Beispiel



Algorithmus zur topologischen Nummerierung:

while $V \neq \emptyset$ **do**

nummeriere eine Quelle mit der nächsten Nummer

streiche diese Quelle aus V

od

Zusammenhang

Definition

Ein Digraph heißt *zusammenhängend, wenn der zugrundeliegende ungerichtete Graph zusammenhängend ist.*

Starke Zusammenhangskomponenten

Definition

Sei $G = (V, A)$ ein Digraph. Man definiert eine Äquivalenzrelation $R \subseteq V \times V$ wie folgt:

$$uRv \iff \begin{cases} \text{es gibt in } G \text{ einen gerichteten Pfad von } u \text{ nach } v \\ \text{und einen gerichteten Pfad von } v \text{ nach } u. \end{cases}$$

Die von den Äquivalenzklassen dieser Relation induzierten Teilgraphen heißen die *starken Zusammenhangskomponenten von* G.

Starke Zusammenhangskomponenten

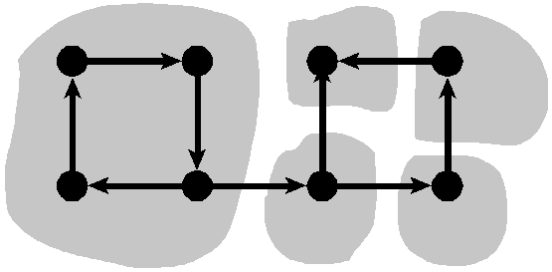
Definition

Sei $G = (V, A)$ ein Digraph. Man definiert eine Äquivalenzrelation $R \subseteq V \times V$ wie folgt:

$$uRv \iff \begin{cases} \text{es gibt in } G \text{ einen gerichteten Pfad von } u \text{ nach } v \\ \text{und einen gerichteten Pfad von } v \text{ nach } u. \end{cases}$$

Die von den Äquivalenzklassen dieser Relation induzierten Teilgraphen heißen die *starken Zusammenhangskomponenten von* G.

Beispiel



Durchsuchen von Graphen

Gesucht sind Prozeduren, die alle Knoten (eventuell auch alle Kanten) mindestens einmal besuchen und möglichst effizient sind.

Tiefensuche, *depth first search*

Sei $G = (V, E)$ ein ungerichteter Graph, gegeben als Adjazenzliste.

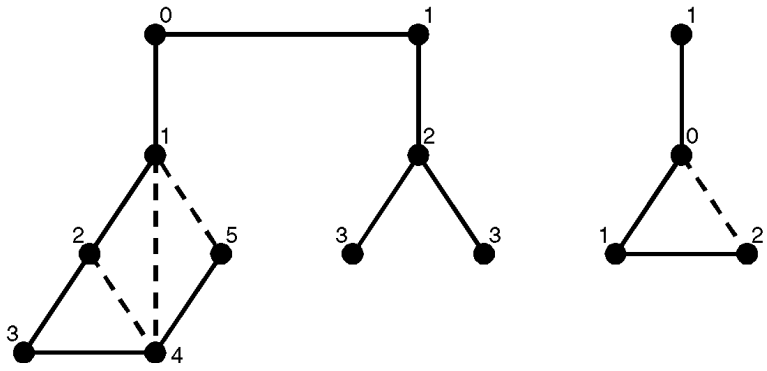
```
algorithm DFS
  void proc DFSvisit(node v)
    visited[v] := true
    pre[v] := ++precount
    for all u ∈ adjacency_list[v] do
      if not visited[u] then
        type[(v,u)] := 'Baumkante'
        parent[u] := v
        DFSlevel[u] := DFSlevel[v]+1
        DFSvisit(u)
      elsif u ≠ parent[v] then
        type[(v,u)] := 'Rückwärtskante'
      fi
    od
    post[v] := ++postcount
  end proc
```

Fortsetzung

```
co Initialisierung: oc
for all v ∈ V do
  visited[v] := false
  pre[v] := post[v] := 0
od
precount := postcount := 0
for all v ∈ V do
  if not visited[v] then
    DFSlevel[v] := 0
    parent[v] := null
    DFSvisit(v)
  fi
od
end
```

Beispiel (gestrichelt sind Rückwärtskanten)

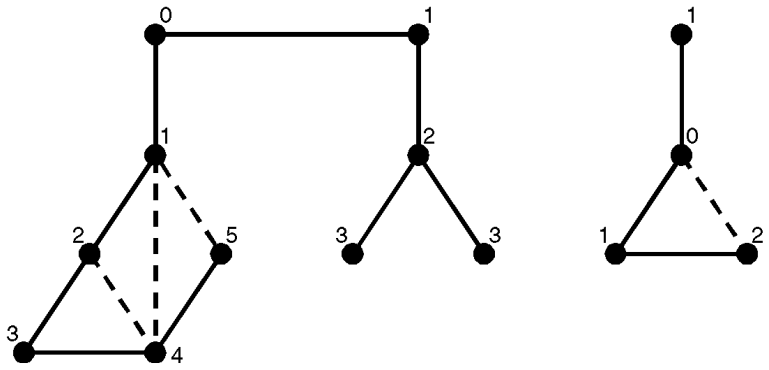
DFS-Level:



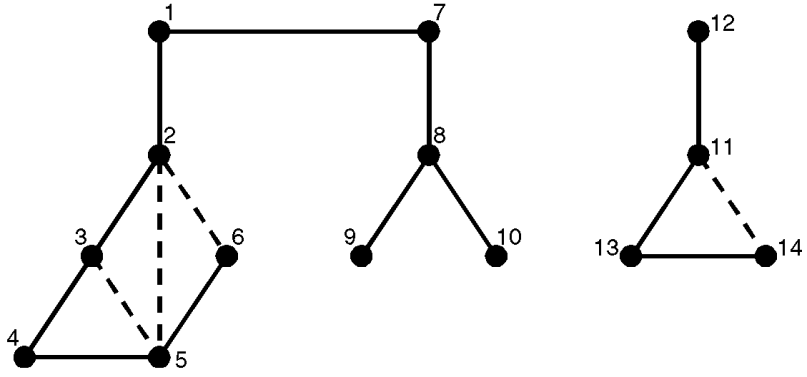
Präorder-Nummer:

Beispiel (gestrichelt sind Rückwärtskanten)

DFS-Level:

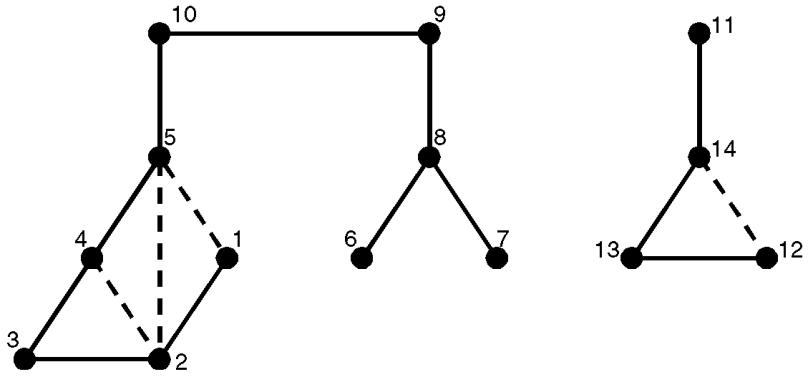


Präorder-Nummer:



Beispiel (Fortsetzung)

Postorder-Nummer:



Beobachtung: Die Tiefensuche konstruiert einen Spannwald des Graphen. Die Anzahl der Bäume entspricht der Anzahl der Zusammenhangskomponenten von G.

Satz

Der Zeitbedarf für die Tiefensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V| + |E|\big) \; .$$

Beweis.

Aus Algorithmus ersichtlich. □

Beobachtung: Die Tiefensuche konstruiert einen Spannwald des Graphen. Die Anzahl der Bäume entspricht der Anzahl der Zusammenhangskomponenten von G.

Satz

Der Zeitbedarf für die Tiefensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V| + |E|\big) \; .$$

Beweis.

Aus Algorithmus ersichtlich. □

Beobachtung: Die Tiefensuche konstruiert einen Spannwald des Graphen. Die Anzahl der Bäume entspricht der Anzahl der Zusammenhangskomponenten von G.

Satz

Der Zeitbedarf für die Tiefensuche ist (bei Verwendung von Adjazenzlisten)

$$O\big(|V| + |E|\big) \ .$$

Beweis.

Aus Algorithmus ersichtlich. □

Tiefensuche im Digraphen: Für gerichtete Graphen verwendet man obigen Algorithmus, wobei man die Zeilen

```
elsif u ≠ parent[v] then
  type[(v,u)] := 'Rückwärtskante'
fi
```

ersetzt durch

```
elsif pre[u] > pre[v] then
  type[(v,u)] := 'Vorwärtskante'
elsif post[u] ≠ 0 then
  type[(v,u)] := 'Querkante'
else
  type[(v,u)] := 'Rückwärtskante'
fi
```

Beispiel (Präorder-Nummer)

[4ex]