6 Tiefensuche in ungerichteten Graphen: Zweifache Zusammenhangskomponenten

Der Algorithmus ist ganz genau derselbe wie im gerichteten Fall!
Abbildung 1 zeigt noch einmal den gerichtete Fall und danach betrachten wir in Abbildung 2 den ungerichteten Fall auf dem analogen Graphen (Kanten " ——>—— " sind Baumkanten, über die entdeckt wird.)

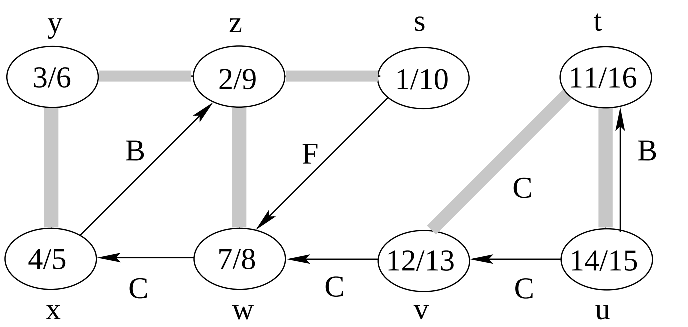


Abbildung 1: Das Ergebnis der Tiefensuche auf einem gerichteten Graphen

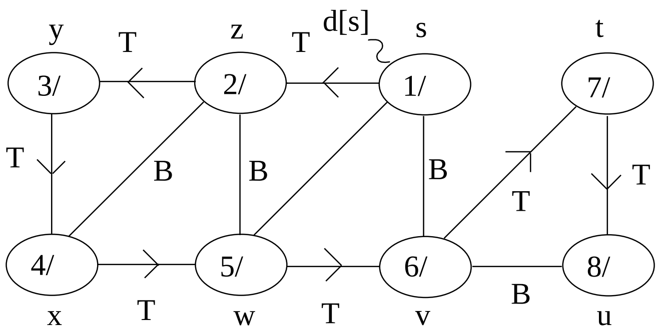
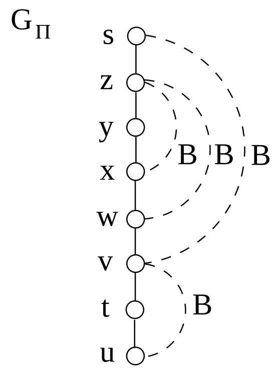


Abbildung 2: vorhergender Graph, ungerichtete Variante

Im ungerichteten Graphen gilt: beim ersten Gang durch eine Kante stößt man

- auf einen weißen Knoten (Kante: T)
- auf einen grauen Knoten (Kante: B)
- <u>nicht</u> auf einen schwarzen Knoten (Kante: F oder C)

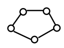
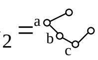


Π [z] = s, Π [y] = z, ...

Im ungerichteten Fall der Tiefensuche ist festzuhalten:

- Jede Kante wird zweimal betrachtet: $\{u, v\}$ bei DFS-visit(u) und bei DFS-visit(v).
- Maßgeblich für den Kantentyp (Baumkante, Rückwärts-, Vorwärts-, Kreuzkante) ist die Betrachtung:
  - (i) $\{u, v\}$ Baumkante $\Longleftrightarrow$ Beim ersten Betrachten von $\{u, v\}$ findet sich ein weißer Knoten.
  - (ii) $\{u, v\}$ Rückwärtskante $\Longleftrightarrow$ Beim ersten Gehen von $\{u, v\}$ findet sich ein bereits grauer Knoten
  - (iii) Beim ersten Gehen kann sich kein schwarzer Knoten finden. Deshalb gibt es weder Kreuz- noch Vorwärtskanten.

Der Weiße-Weg-Satz gilt hier vollkommen analog.
Kreise erkennen ist ganz ebenfalls analog mit Tiefensuche möglich.
Der Begriff der starken Zusammenhangskomponente ist nicht sinnvoll, da Weg (u, v) im ungerichteten Fall ebenso (v, u) ist.

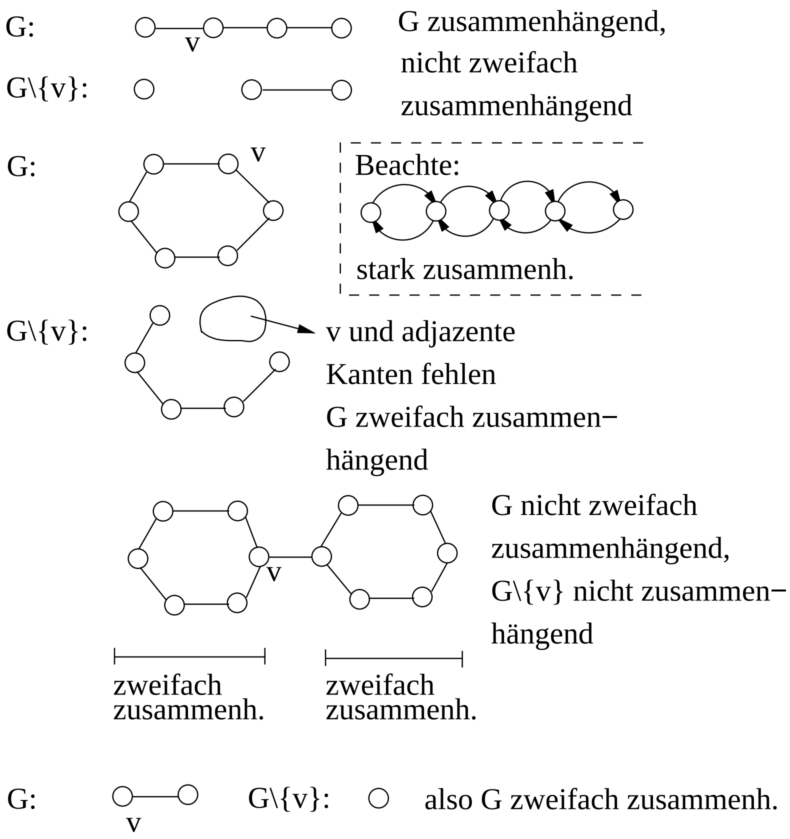
Stattdessen: *zweifach zusammenhängend.* Unterschied zwischen $G_1 =$  und $G_2 =$ 

?
Löschen wir in G_1 einen beliebigen Knoten, hängt der Rest noch zusammen. Für a, b, c im Graphen G_2 gilt dies nicht!
Für den Rest dieses Kapitels gilt nun folgende *Konvention*:
Ab jetzt gehen wir nur immer von zusammenhängenden, gerichteten Graphen aus.

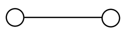
Definition 6.1(zweifach zusammenhängend): $G\backslash\{v\}$ *ist zweifach zusammenhängend* $\Leftrightarrow$ G *ist zusammenhängend für alle* $v \in V$.
$G \setminus \{v\} = (V \setminus \{v\}, E \setminus \{\{u, v\} | u \in V\})$.

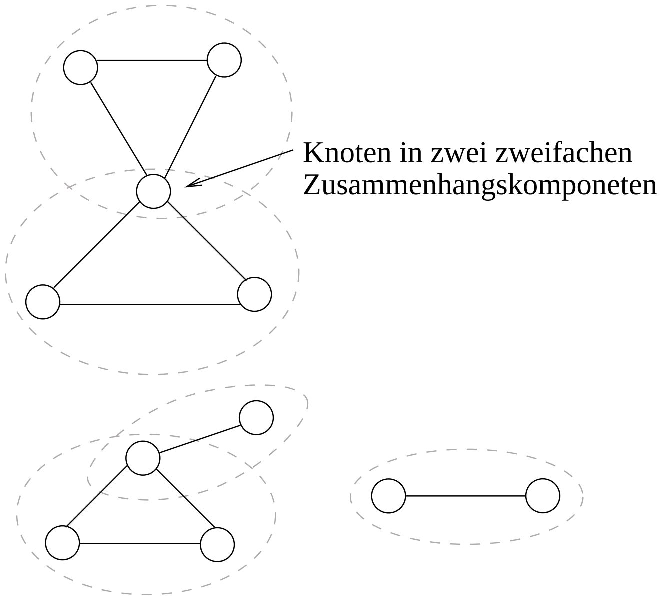
Was ist die Bedeutung zweifach zusammenhängend?

Satz 6.1(Menger 1927): *G zweifach zusammenh. $\Longleftrightarrow$ Für alle $u, v \in V, u \neq v$ gibt es zwei disjunkte Wege zwischen u und v in G.*



Das heißt, Wege $(u, u_1, u2, \dots, u_m, v)$, $(u, u_1', u_2', \dots, u_m', v)$ *mit* $\{u_1, \dots, u_m\} \cap \{u_1', \dots, u_m'\} = \emptyset$ $\square$

Beweis erfolgt auf überraschende Weise später.
Ein induktiver Beweis im Buch Graphentheorie von Reinhard Diestel. Die Richtung „$\Leftarrow$“ ist einfach. Beachte noch, ist u  v, so tut es der Weg (u, v), mit leerer Menge von Zwischenknoten (also nur 1 Weg).
Nun gilt es wiederm nicht zweifach zusammenhängende Graphen in seine zweifach zusammenhängenden Bestandteile zu zerlegen, in die zweifachen (Zusammenhangs-)Komponenten. **Beispiel 6.1**:



Formal

Definition 6.2(zweifache Komponenten): *Ein Teilgraph $H = (W, F)$ von G ist eine zweifache Komponente $\iff$ H ist ein maximaler zweifach zusammenhängender Teilgraph von G (maximal bezüglich Knoten und Kanten).* □

(a) Jede Kante ist in genau einer zweifachen Komponente.

(b) Sind $H_1 = (W_1, F_1), \ldots, H_k = (W_k, F_k)$ die zweifachen Komponeten von $G = (V, E)$, so ist $F_1, \ldots, F_k$ eine Partition (Einteilung) von E.

$F_i \cap F_j = \emptyset$ für $i \neq j, F_1 \cup \cdots \cup F_k = E$.

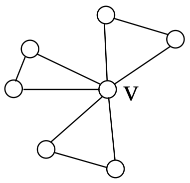
Beweis.

(a) Sei also $H_1 \neq H_2$ zweifach zusammenhängend, dann ist $H = H_1 \cup H_2$ zweifach zusammenhängend:
Für w aus $H_1, w \neq u, w \neq v$ ist $H \backslash \{w\}$ zusammenhängend (wegen Maximalität). Ebenso w aus H_2. Für $w = u$ ist, da immer noch v da ist, $H \setminus \{w\}$ zweifach zusammenhängend. Also wegen Maximalität zweifache Komponenten $\supseteq H_1 \cup H_2$.

(b) $F_i \cap F_j = \emptyset$ wegen (a). Da Kante (u, v) zweifach zusammenhängend ist, ist nach Definition jede Kante von E in einem F_1.

□

Bei Knoten gilt (a) oben nicht:



Bezeichnung:
Eine Kante, die eine zweifache Komponente ist, heißt Brückenkante

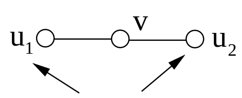
v gehört zu 3 zweifachen Komponenten. v ist ein typischer Artikulationspunkt.

Definition 6.3(Artikulationspunkt): *v ist Artikulationspunkt von G* $\iff G\setminus\{v\}$ *nicht zusammenhängend.*

Bemerkung 6.2:

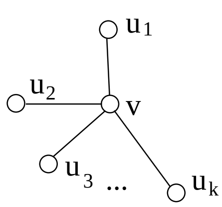
v ist Artikulationspunkt $\iff$ v gehört zu ≥ 2 zweifachen Komponenten

Beweis. „$\Rightarrow$" Ist v Artikulationspunkt. Dann gibt es Knoten $u, w, u \neq v, w \neq v$, so dass jeder Weg von u nach w von der Art $(u, \ldots, v, \ldots, w)$ ist. (Sonst $G \setminus \{v\}$ zusammenhängend)
Also u, w nicht in einer zweifachen Komponente. Dann ist jeder Weg von der Art $(u, \ldots, u_1, v, u_2, \ldots, 2)$, so dass u_1, u_2 nicht in einer zweifachen Komponente sind. Sonst ist $G \setminus \{v\}$ Weg $(u, \ldots, u_1, u_2, \ldots w)$ (ohne v), Widerspruch. Also haben wir



verschiedene Komponenten.

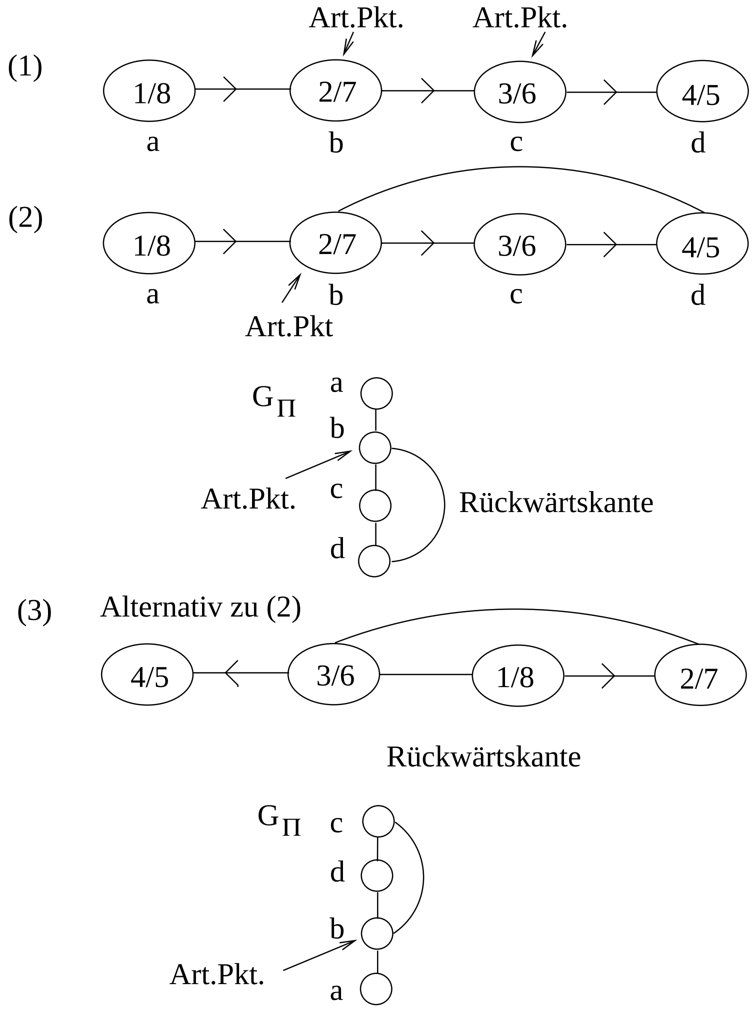
Die Kante $\{u_1, v\}$ gehört zu einer Komponente (eventuell ist sie selber eine), ebenso $\{v, u_2\}$. Die Komponenten der Kanten sind verschieden, da u_1, u_2 in verschiedenen Komponenten liegen. Alos v liegt in den beiden Komponenten. „$\Leftarrow$" Gehört also v zu > 2 Komponenten. Dann



und ≥ 2 Kanten von $\{u_i, v\}$ in verschiedenen Komponenten. Etwa u_1, u_2. Aber u_1 und u_2 in 2 verschiedenen Komponenten. Also gibt es w, so dass in $G \setminus \{w\}$ kein

Weg $u_1 \circ\!\!-\!\!-\!\!\circ u_2$ existiert. Denn wegen $u_1 \circ\!\!-\!\!-\!\!\overset{v}{\circ}\!\!-\!\!-\!\!\circ u_2$ muss $w = v$ sein und v Artikulationspunkt. $\square$

Graphentheoretische Beweise sind nicht ganz so leicht!
Artikulationspunkte und Tiefensuche?



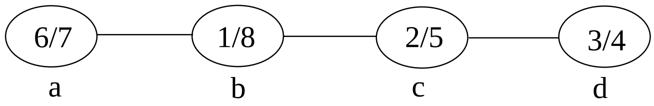
Was haben die Artikulationspunkte in allen genannten Fällen gemeinsam? Der Artikulationspunkt (sofern nicht Wurzel von G_Π) hat einen Sohn in G_Π, so dass es von dem Sohn oder Nachfolger keine Rückwärtskante zum echten Vorgänger gibt!

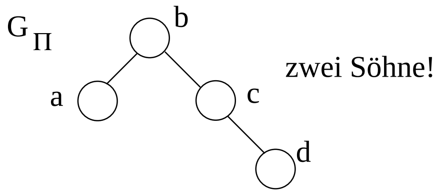
(1) Gilt bei b, c. d ist kein Artikulationspunkt und das Kriterium gilt auch nicht.

(2) b ist Artikulationspunkt, Sohn c erfüllt das Kriterium. c, d erfüllen es nicht, sind auch keine Artikulationspunkte.

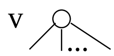
(3) Hier zeigt der Sohn a von b an, dass b ein Artikulationspunkt ist. Also keineswegs immer derselbe Sohn!

Was ist, wenn der Artikulationspunkt Wurzel von G_Π ist?

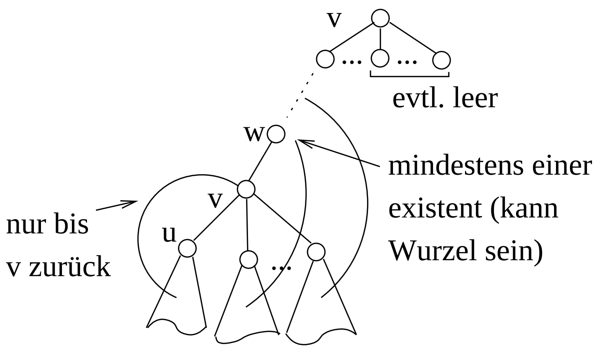




Satz 6.2(Artikulationspunkte erkennen): *Sei v ein Knoten von G und sei eine Tiefensuche gelaufen. (Erinnerung: G immer zusammenhängend).*

a) Ist v Wurzel von G_Π. v ist Arikulationspunkt $\iff$ G_Π  *$\leq$ 2Söhne.*

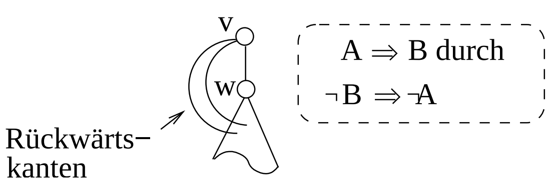
b) Ist v nicht Wurzel von G_Π.
v ist Artikulationspunkt
$\iff$



Das heißt, v hat einen Sohn (hier u), so dass von dem und allen Nachfolgern in G_Pi keine (Rückwärts-)Kanten zu echtem Vorgänger von v.

Beweis.

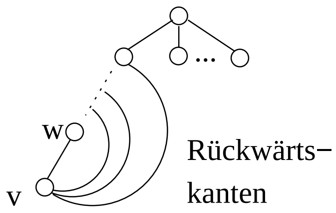
a) „$\Leftarrow$“ $G \setminus \{v\}$ nicht zusammenhängend, damit v Artikulationspunkt. „$\Rightarrow$ “ Hat v nur einen Sohn, dann



dann $G \setminus \{v\}$ zusammenhängend. (Können in $G \setminus \{v\}$ über w statt v gehen.) Also ist v kein Artikulationspunkte. Ist v ohne Sohn, dann ist er kein Artikulationspunkt.

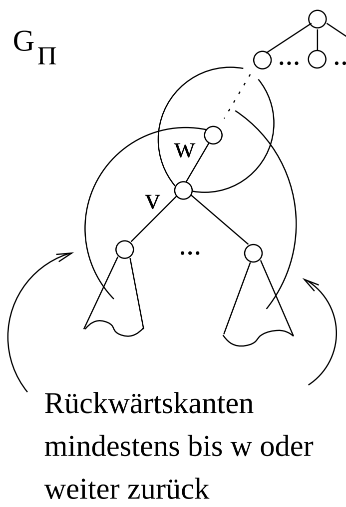
b) „$\Longleftarrow$ “ In $G\{v\}$ kein Weg u ○——○ w, also ist v Artikulationspunkt.
„$\Rightarrow$ “ Gelte die Behauptung nicht, also v hat keinen Sohn wie u in G_Π.

1. Fall v hat keinen Sohn. Dann in G_Π



In $G \setminus \{v\}$ fehlen die Rückwärtskanten und $\{w, v\}$. Also bleibt der Rest zusammenhängend. v ist kein Artikulationspunkt.

2. Fall v hat Söhne, aber keinen wie u in der Behauptung
Dann:



In $G \setminus \{v\}$ bleiben die eingetragenen Rückwärtskanten, die nicht mit v inzident sind, stehen. Also ist $G \setminus \{v\}$ zusammenhängend. Also ist v kein Artikulationspunkt

$$\boxed{\begin{array}{c} A \Rightarrow B \text{ durch} \\ \neg B \Rightarrow \neg A. \end{array}}$$

□

Wie kann man Artikulationspunkte berechnen?
Wir müssen für alle Söhne in G_Π wissen, wie weit es von dort aus zurück geht.

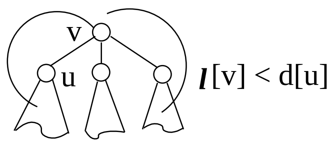
Definition 6.4: *Sei DFS(G) gelaufen, also G_Π vorliegend.*

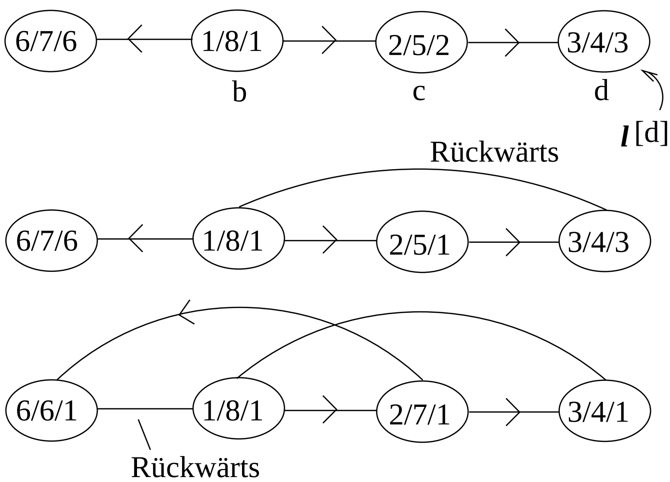
a) *Für Knoten v ist die Menge von Knoten $L(v)$ gegeben durch $w \in L(u) \iff w = v$ oder w Vorgänger von v und es gibt eine Rückwärtskante von v oder dem Nachfolger zu w.*

b) *$l[v] = Min\{d[w] | w \in L(v)\}$ ist der Low-Wert von v. ($l[v]$ hängt von Lauf von DFS(G) ab.)*

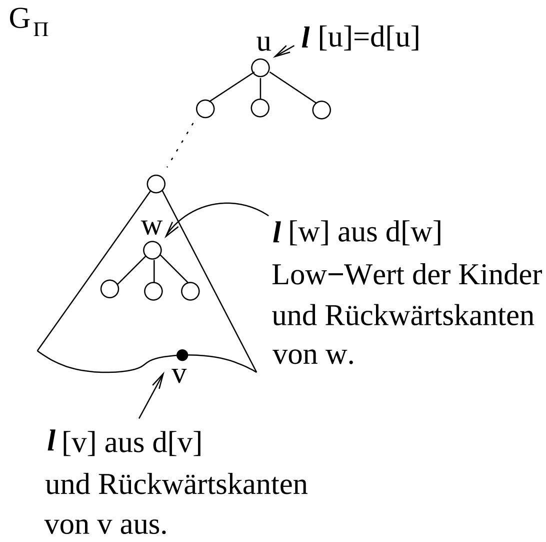
Folgerung 6.1: *Sei DFS(G) gelaufen und v nicht Wurzel von G_Π.*
v ist Artikulationspunkt $\iff$ v hat Sohn u in G_Π mit $l[u] \geq d[v]$.

Beachte: $l[v] = d[v] \Rightarrow v$ Artikulationspunkt. Keine Äquivalenz.





6.1 Berechnung von l[v]



Korrektheit: Induktion über die Tiefe des Teilbaumes 

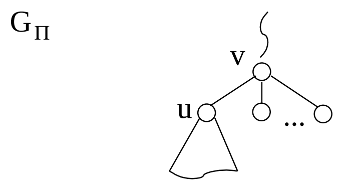
6.2 Algorithmus (l-Werte)

Modifikation von DFS-visit(u):

```
MDFS-visit(u)                        //hier col[u] = weiß
⋮
l[u]=d[u]
⋮
for each v ∈ Adj[u] do{
   if (col[v] == weiß){
      Π[v] = u; MDFS-visit(v);
      l[u] = MIN{l[u], l[v]}
   }
   if (col[v] == grau) and (Π[u] ≠ v){
      l[u] = MIN{l[u], d[v]}         //d[v]! nicht l[v], keine Iteration.
   }
}
```

Damit können wir Artikulationspunkte in Linearzeit erkennen. Es bleiben die zweifachen Komponenten zu finden. Seien die l-Werte gegeben. Nun 2. Tiefensuche folgendermaßen:

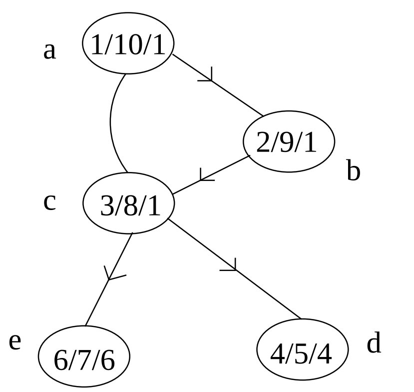
- Knoten, der entdeckt wird, auf einem Keller speichern.



Sei jetzt $l[u] \leq d[v]$.

- Am Ende von DFS-visit(u) Keller bis v(!) ausgeben. Ist zweifache Komponente. v wieder oben drauf schreiben.

Beispiel 6.3:



Kellerinhalt:

			↓
		↓	a
	↓	b	a
↓	c	b	a
d	c	b	a

$l[d] \geq d[c]$ Ausgabe d, c

	c	b	a
e	c	b	a

Ausgabe e, c

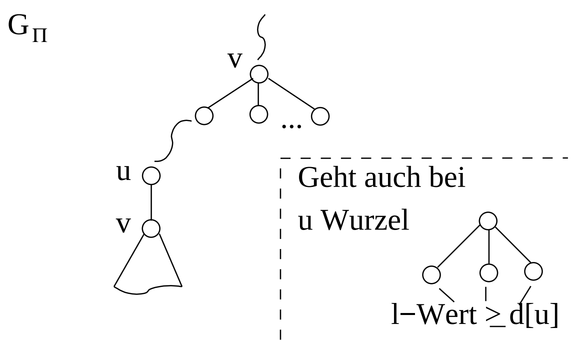
c b a

Ausgabe c, b, a am Ende.

```
DFS(G);                                      /* modifiziert für l-Werte */
DFS(G);    /* in derselben(!) Reihenfolge wie 1. mit col[u]=w */
NDFS-visit(u);
⋮ foreach v∈Adj[u] do
    if col[v]==w then
        Pi[v]=u, v auf Keller;
        NDFS-visit(v);
        if l[v]≥d[u] then
            Ausgabe bis v inklusive u noch hinzu ausgeben;
        end
    end
end
```

6.3 Algorithmus (Zweifache Komponenten)

Beweis. Korrektheit induktiv über $s = \#$ Zweifache Komponenten von G.
Induktionsanfang: $s = 1$
Ausgabe nur am Ende, da kein Artikulationspunkt. Also korrekt. Induktionsschluss
Gelte Behauptung für alle(!) Graphen mit $s \geq 1$ zweifachen Komponenten. Zeige dies für G mit $s + 1$ Komponenten.
Wir betrachten die 2. Tiefensuche des Algorithmus. Die l-Werte stimmen also bereits, wegen der Koorektheit der 1. Tiefensuche.
Wir betrachten den Baum G_Π, welcher der gleiche wie bei der 1. Tiefensuche und 2. Tiefensuche ist.



Sei NDFS-visit(v) der erste Aufruf, nach dm die Ausgabe erfolgt. Dann ist $l[v] \geq [u]$.
Ist l[v] = d[v], dann ist v Blatt (sonst vorher Ausgabe) und v, u wird ausgegeben und u kommt auf die Kellerspitze.
Ist l[v] = d[u], dann Ausgabe des Kellers bis u und u kommt wieder auf die Kellerspitze.
Damit wird eine zweifache Komponente ausgegeben. (Mehr kann nicht dazu gehören, weniger nicht, da bisher kein Artikulationspunkt). Nach Ausgabe der Komponente

liegt eine Situation vor, die in DFS-visit(u) auftritt, wenn wir den Graphen betrachten, in dem $\{u, v\}$ und die anderen ausgegegenen Knoten gelöscht sind.
Auf diesem ist die Induktionsvoraussetzung anwendbar und der Rest wird richtig ausgegeben. $\square$