Tiefensuche

Theorem

Gegeben sei ein gerichteter Graph $G = (V, E)$ (in Adjazenzlistendarstellung). Durch Tiefensuche kann ein DFS-Wald inklusive der Funktionen $d : V \to \mathbf{N}$, $f : V \to \mathbf{N}$ in $O(|V| + |E|)$ Schritten (also in linearer Zeit) berechnet werden.

Beweis.

(Skizze) Solange ein Knoten grau ist, wird jede inzidente Kante einmal besucht. Jeder Knoten wechselt seine Farbe nur zweimal, jedesmal mit konstantem Aufwand. Jede Kante wird daher ebenfalls nur einmal besucht. $\square$

Zusammenhangskomponenten

Definition

Es sei $G = (V, E)$ ein ungerichteter Graph. Ein Pfad der Länge k von u_1 nach u_{k+1} ist eine Folge $(u_1, u_2), (u_2, u_3), \ldots, (u_k, u_{k+1})$ von Kanten aus E wobei $u_1, \ldots, u_{k+1}$ paarweise verschieden sind.

Wir sagen u und v sind zusammenhängend, wenn es einen Pfad von u nach v gibt.

Eine Menge $C \subseteq V$ ist eine Zusammenhangskomponente, wenn alle Knoten in C zusammenhängend sind und es keine echte Obermenge von C mit dieser Eigenschaft gibt.
(Alternativ: Die Zusammenhangskomponenten sind die Äquivalenzklassen der Relation „zusammenhängend".)

Zusammenhangskomponenten

Definition

Es sei $G = (V, E)$ ein ungerichteter Graph. Ein Pfad der Länge k von u_1 nach u_{k+1} ist eine Folge $(u_1, u_2), (u_2, u_3), \ldots, (u_k, u_{k+1})$ von Kanten aus E wobei $u_1, \ldots, u_{k+1}$ paarweise verschieden sind.

Wir sagen u und v sind zusammenhängend, wenn es einen Pfad von u nach v gibt.

Eine Menge $C \subseteq V$ ist eine Zusammenhangskomponente, wenn alle Knoten in C zusammenhängend sind und es keine echte Obermenge von C mit dieser Eigenschaft gibt.
(Alternativ: Die Zusammenhangskomponenten sind die Äquivalenzklassen der Relation „zusammenhängend".)

Zusammenhangskomponenten

Definition

Es sei $G = (V, E)$ ein ungerichteter Graph. Ein Pfad der Länge k von u_1 nach u_{k+1} ist eine Folge $(u_1, u_2), (u_2, u_3), \ldots, (u_k, u_{k+1})$ von Kanten aus E wobei $u_1, \ldots, u_{k+1}$ paarweise verschieden sind.

Wir sagen u und v sind zusammenhängend, wenn es einen Pfad von u nach v gibt.

Eine Menge $C \subseteq V$ ist eine Zusammenhangskomponente, wenn alle Knoten in C zusammenhängend sind und es keine echte Obermenge von C mit dieser Eigenschaft gibt.
(Alternativ: Die Zusammenhangskomponenten sind die Äquivalenzklassen der Relation „zusammenhängend“.)

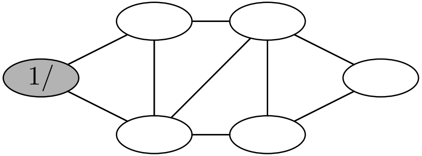
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



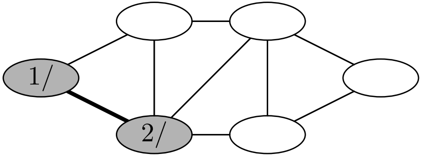
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



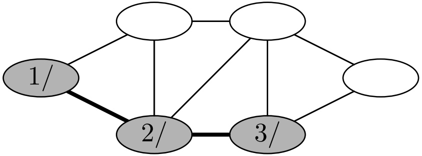
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



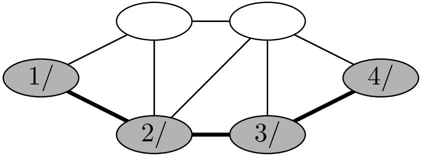
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



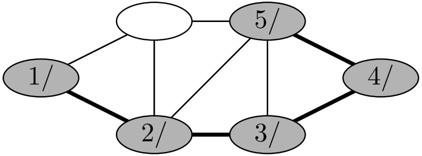
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



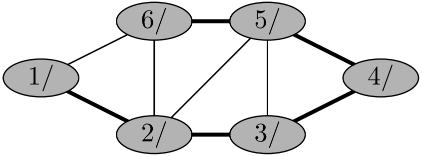
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



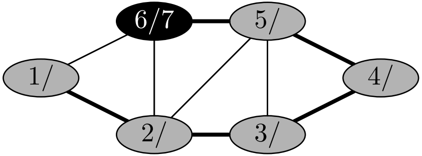
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



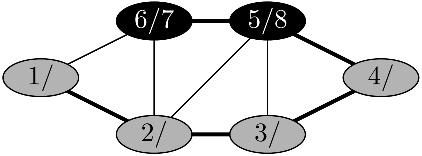
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



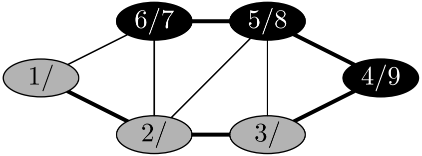
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



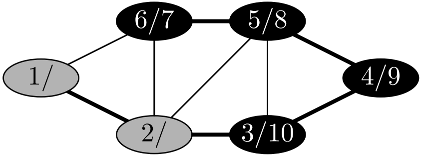
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



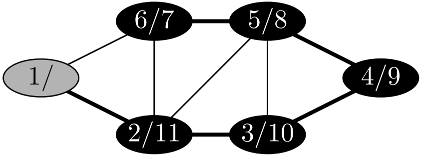
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



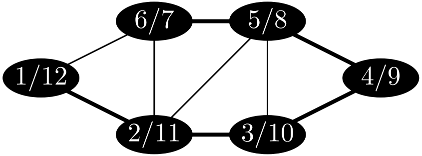
Zusammenhangskomponenten

Theorem

Die Zusammenhangskomponenten eines ungerichteten Graphen können in linearer Zeit gefunden werden.

Beweis.

Die Knoten, die jeweils in einem Aufruf der Tiefensuche schwarz werden, gehören zu einer Komponente. □



Finden von Kreisen

Theorem

Gegeben sei ein gerichteter Graph G. Dann können wir in linearer Zeit feststellen, ob G azyklisch ist (keine Kreise enthält).

Beweis.

Führe eine Tiefensuche auf G aus.

Behauptung: G ist genau dann azyklisch, wenn es keine Rückwärtskanten gibt.

$\Rightarrow$ Wenn es eine Rückwärtskante von u nach v gibt, dann gibt es auch einen Pfad von v nach u im DFS-Wald. Dies ist ein Kreis.

$\Leftarrow$ Angenommen es gibt einen Kreis. Sei u der Knoten auf dem Kreis mit minimalem $d(u)$ und v der Knoten auf dem Kreis vor u. Dann gilt $d(u) < d(v)$ und $f(v) < d(u)$. Also ist (v, u) eine Rückwärtskante. $\square$

Finden von Kreisen

Theorem

Gegeben sei ein gerichteter Graph G. Dann können wir in linearer Zeit feststellen, ob G azyklisch ist (keine Kreise enthält).

Beweis.

Führe eine Tiefensuche auf G aus.

Behauptung: G ist genau dann azyklisch, wenn es keine Rückwärtskanten gibt.

$\Rightarrow$ Wenn es eine Rückwärtskante von u nach v gibt, dann gibt es auch einen Pfad von v nach u im DFS-Wald. Dies ist ein Kreis.

$\Leftarrow$ Angenommen es gibt einen Kreis. Sei u der Knoten auf dem Kreis mit minimalem $d(u)$ und v der Knoten auf dem Kreis vor u. Dann gilt $d(u) < d(v)$ und $f(v) < d(u)$. Also ist (v, u) eine Rückwärtskante. $\square$

Finden von Kreisen

Theorem

Gegeben sei ein gerichteter Graph G. Dann können wir in linearer Zeit feststellen, ob G azyklisch ist (keine Kreise enthält).

Beweis.

Führe eine Tiefensuche auf G aus.

Behauptung: G ist genau dann azyklisch, wenn es keine Rückwärtskanten gibt.

$\Rightarrow$ Wenn es eine Rückwärtskante von u nach v gibt, dann gibt es auch einen Pfad von v nach u im DFS-Wald. Dies ist ein Kreis.

$\Leftarrow$ Angenommen es gibt einen Kreis. Sei u der Knoten auf dem Kreis mit minimalem $d(u)$ und v der Knoten auf dem Kreis vor u. Dann gilt $d(u) < d(v)$ und $f(v) < d(u)$. Also ist (v, u) eine Rückwärtskante. $\square$

Finden von Kreisen

Theorem

Gegeben sei ein gerichteter Graph G. Dann können wir in linearer Zeit feststellen, ob G azyklisch ist (keine Kreise enthält).

Beweis.

Führe eine Tiefensuche auf G aus.
Behauptung: G ist genau dann azyklisch, wenn es keine Rückwärtskanten gibt.
$\Rightarrow$ Wenn es eine Rückwärtskante von u nach v gibt, dann gibt es auch einen Pfad von v nach u im DFS-Wald. Dies ist ein Kreis.
$\Leftarrow$ Angenommen es gibt einen Kreis. Sei u der Knoten auf dem Kreis mit minimalem $d(u)$ und v der Knoten auf dem Kreis vor u. Dann gilt $d(u) < d(v)$ und $f(v) < d(u)$. Also ist (v, u) eine Rückwärtskante. $\square$

Übersicht

Starke Zusammenhangskomponenten

Definition

Es sei $G = (V, E)$ ein gerichteter Graph. Ein Pfad der Länge k von u_1 nach u_{k+1} ist eine Folge $(u_1, u_2), (u_2, u_3), \dots, (u_k, u_{k+1})$ von Kanten aus E wobei $u_1, \dots, u_{k+1}$ paarweise verschieden sind.

Eine Menge $C \subseteq V$ ist eine starke Zusammenhangskomponente, wenn es Pfade zwischen allen Paaren von Knoten in C gibt und es keine echte Obermenge von C mit dieser Eigenschaft gibt.

Die starken Komponenten sind eine Verfeinerung der Zusammenhangskomponenten des entsprechenden ungerichteten Graphen.

Starke Zusammenhangskomponenten

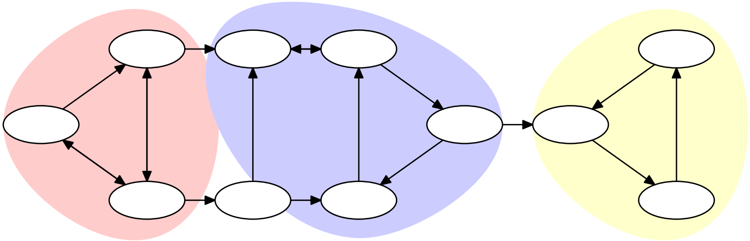
Definition

Es sei $G = (V, E)$ ein gerichteter Graph. Ein Pfad der Länge k von u_1 nach u_{k+1} ist eine Folge $(u_1, u_2), (u_2, u_3), \dots, (u_k, u_{k+1})$ von Kanten aus E wobei $u_1, \dots, u_{k+1}$ paarweise verschieden sind.

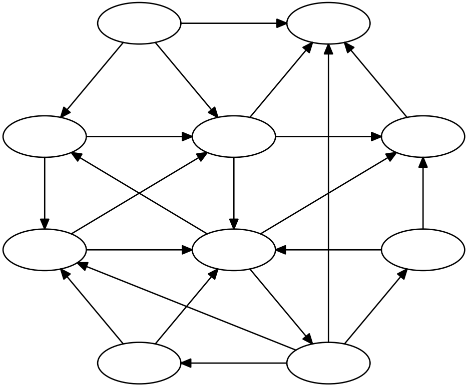
Eine Menge $C \subseteq V$ ist eine starke Zusammenhangskomponente, wenn es Pfade zwischen allen Paaren von Knoten in C gibt und es keine echte Obermenge von C mit dieser Eigenschaft gibt.

Die starken Komponenten sind eine Verfeinerung der Zusammenhangskomponenten des entsprechenden ungerichteten Graphen.

Starke Komponenten – Beispiel



Es gibt genau vier starke Komponenten.

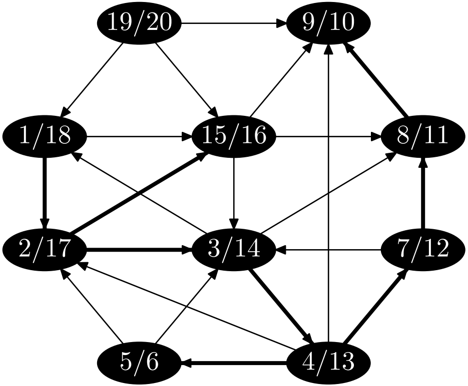


Lemma

Der Knoten mit der größten finish time nach einer DFS ist in einer Quellenkomponente.

Beweis.

Er ist Wurzel eines DFS-Baums T. Wenn es einen anderen DFS-Baum gäbe, von dem eine Kante nach T führte, dann müßte dieser danach besucht werden. □

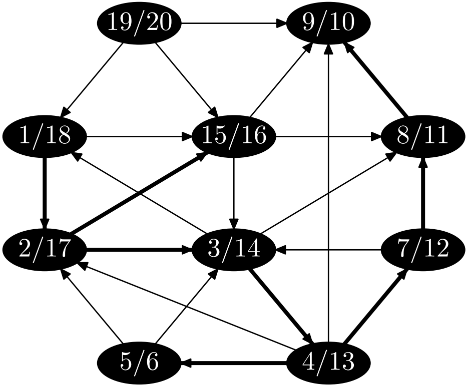


Lemma

Der Knoten mit der größten finish time nach einer DFS ist in einer Quellenkomponente.

Beweis.

Er ist Wurzel eines DFS-Baums T. Wenn es einen anderen DFS-Baum gäbe, von dem eine Kante nach T führte, dann müßte dieser danach besucht werden. □

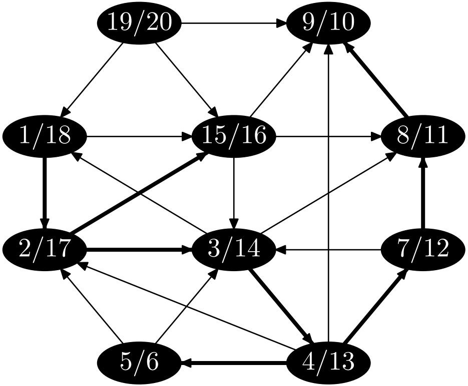


Lemma

Der Knoten mit der größten finish time nach einer DFS ist in einer Quellenkomponente.

Beweis.

Er ist Wurzel eines DFS-Baums T. Wenn es einen anderen DFS-Baum gäbe, von dem eine Kante nach T führte, dann müßte dieser danach besucht werden. □



Quellenkomponente: Starke Komponente, in die keine Kante hineinführt

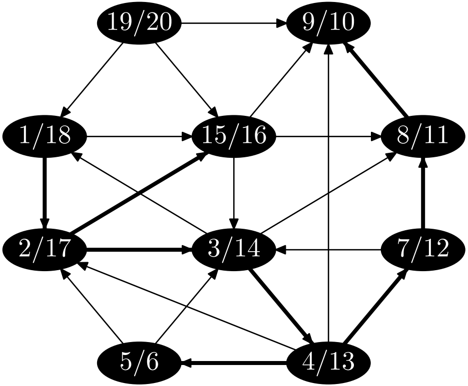
Senkenkomponente: Starke Komponente, aus der keine Kante herausführt

G_r: Wie G, aber Richtung der Kanten umgekehrt

Korollar

Der Knoten mit der größten finish time nach einer DFS in G_r ist in einer Senkenkomponente von G.

Die starken Komponenten von G und G_r sind identisch.



Quellenkomponente: Starke Komponente, in die keine Kante hineinführt

Senkenkomponente: Starke Komponente, aus der keine Kante herausführt

G_r: Wie G, aber Richtung der Kanten umgekehrt

Korollar

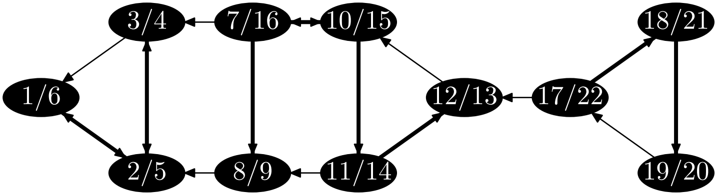
Der Knoten mit der größten finish time nach einer DFS in G_r ist in einer Senkenkomponente von G.

Die starken Komponenten von G und G_r sind identisch.

Starke Komponenten

Lemma

Gegeben sei ein gerichteter Graph G mit einem DFS-Wald.
Der DFS-Wald bleibt samt discover und finish times ein möglicher DFS-Wald, wenn die Quellenkomponente entfernt wird, die den Knoten mit maximaler finish time enthält.

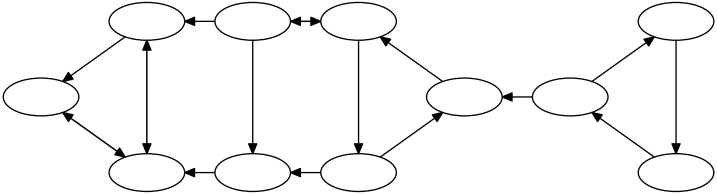


Beweis.

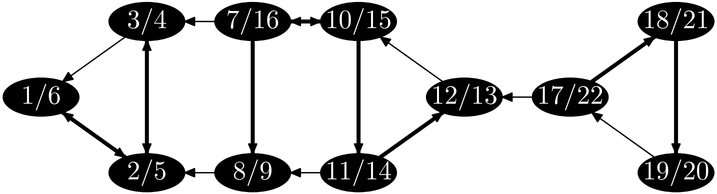
Der Baum spannt die ganze starke Komponente auf. Er wurde als letzter besucht. □

Starke Komponenten – Algorithmus von Kosaraju

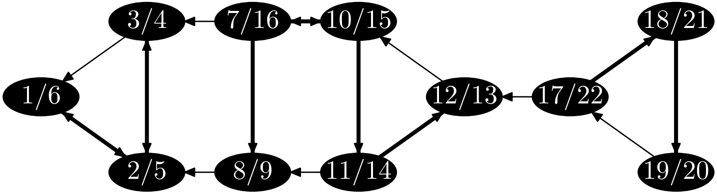
1. Bilde G_r (Kanten umdrehen)
2. Führe Tiefensuche auf G_r aus
3. Ordne die Knoten nach absteigender finish time $f(v)$
4. Führe in dieser Reihenfolge Tiefensuche auf G aus
5. Jeder Baum im DFS-Wald spannt eine starke Komponente auf

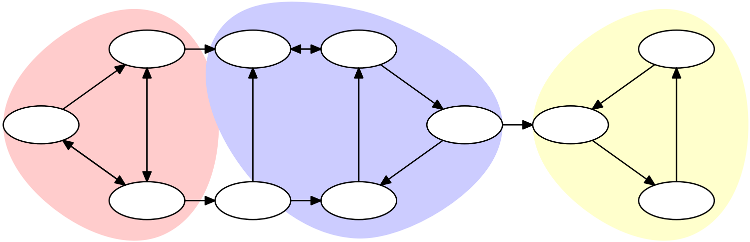


Starke Komponenten – Algorithmus von Kosaraju

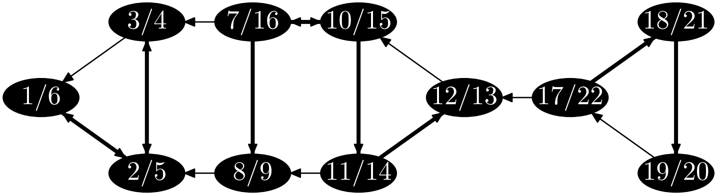


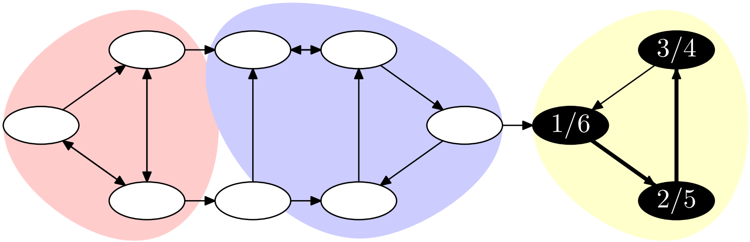
Starke Komponenten – Algorithmus von Kosaraju



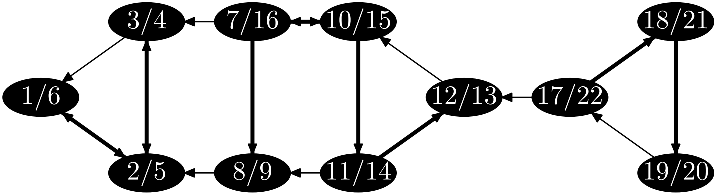


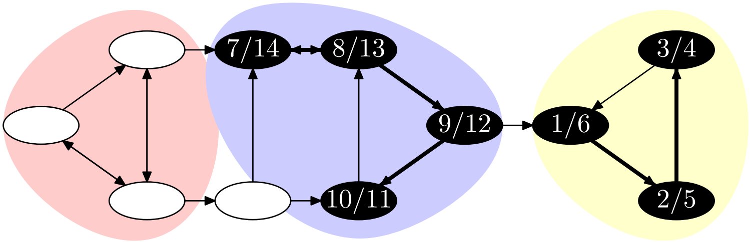
Starke Komponenten – Algorithmus von Kosaraju



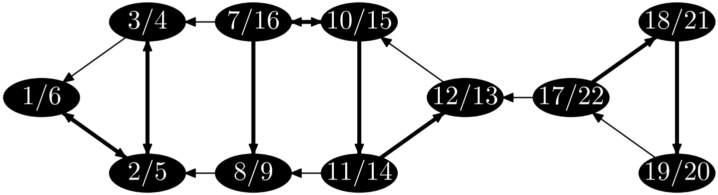


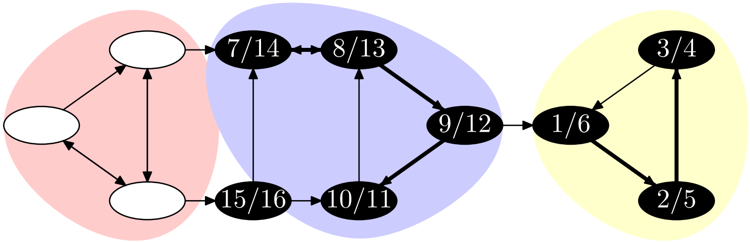
Starke Komponenten – Algorithmus von Kosaraju



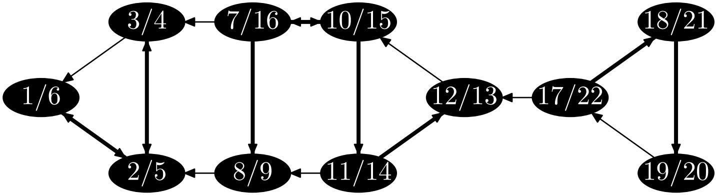


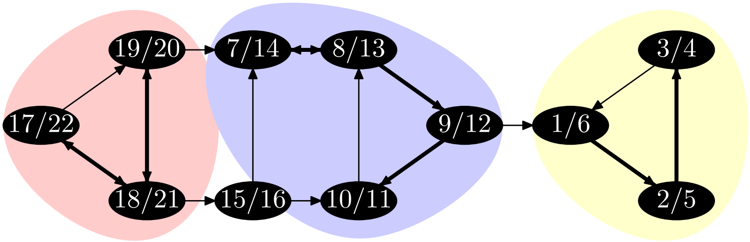
Starke Komponenten – Algorithmus von Kosaraju





Starke Komponenten – Algorithmus von Kosaraju





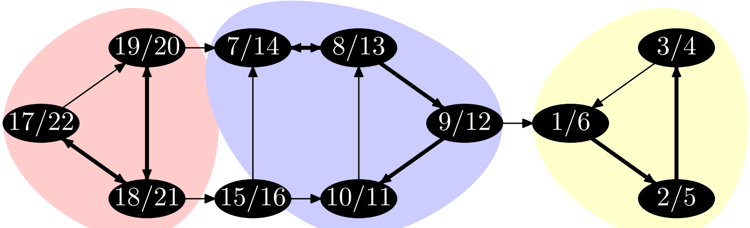
Theorem

Der Algorithmus von Kosaraju berechnet die starken Komponenten eines gerichteten Graphen.

Beweis.

Die DFS auf G beginnt in einem Knoten einer Senkenkomponente von G. Dabei entsteht ein DFS-Baum, der diese starke Komponente aufspannt.

Die DFS fährt dann wieder in einer Senkenkomponente des Restgraphen fort und so weiter. □



Übersicht

Topologisches Sortieren

Ein reflexiv-transitive Hüllen eines DAG entspricht einer Halbordnung.

Definition

Es sei M eine Menge. Wir sagen $\leq \subseteq M \times M$ ist eine Halbordnung auf M, wenn

1. Für alle $x \in M$ gilt $x \leq x$
2. Für alle $x, y \in M$ gilt $x \leq y \wedge y \leq x \Rightarrow x = y$
3. Für alle $x, y, z \in M$ gilt $x \leq y \wedge y \leq z \Rightarrow x \leq z$

Topologisches Sortieren:

Bette eine Halbordnung in eine Ordnung ein.

Ordne die Knoten eines DAG so, daß keine Kante von einem größeren zu einem kleineren Knoten führt.