Planare Graphen - Coloring

Manuel Frohn

RWTH Aachen University, Aachen, Germany

?

Inhaltsverzeichnis

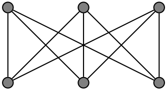
Relevanz

1. Real auftretene Klasse
2. Wichtig für Chip Design und Städteplanung
3. Bedingung für Algorithmen und Sätze

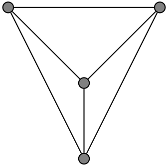
Planare Graphen

Definition

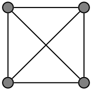
Ein Graph G heißt planar, wenn man in der Lage ist, den Graphen so auf eine Ebene zu zeichnen, dass sich seine Kanten nicht schneiden.



(a) $K_{3,3}$



(b) K_4

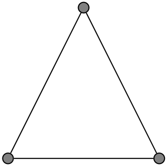


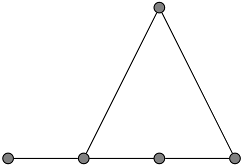
(c) K_4

Minor

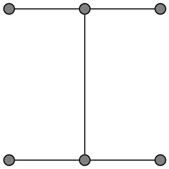
Definition

M heißt Minor von G wenn M aus einem Teilgarphen von G, durch Kantenkontaktion hervorgeht.

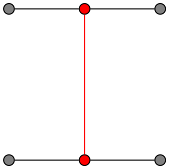




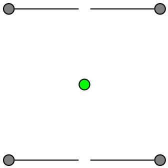
Kantenkontraktion



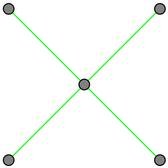
Kantenkontraktion



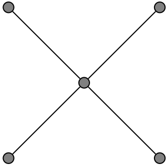
Kantenkontraktion



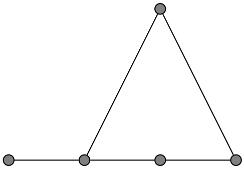
Kantenkontraktion



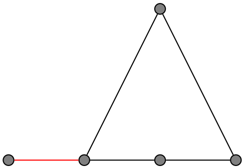
Kantenkontraktion



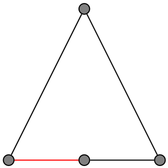
Minor Beispiel



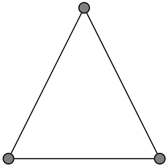
Minor Beispiel



Minor Beispiel



Minor Beispiel



Eulerscher Polyedersatz

Satz

Gegeben ein planarer Graph $G = (V, E)$ *und die Anzahl seiner Gebiete* $|F|$ *gilt:* $|V| - |E| + |F| = 2$

Eulerscher Polyedersatz

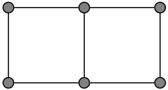
Satz

Gegeben ein planarer Graph $G = (V, E)$ und die Anzahl seiner Gebiete $|F|$ gilt: $|V| - |E| + |F| = 2$

Satz

G *Planar* $\Leftrightarrow |E| \leq 3|V| - 6 \wedge |F| \leq 2|V| - 4$

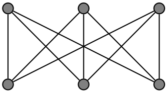
Gebiete



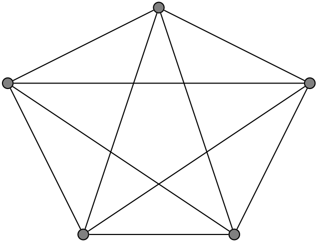
Satz von Kuratowski

Satz

Ein Graph ist genau dann planar, wenn er weder den $K3,3$ noch den K_5 als Minor enthält



(a) $K_{3,3}$



(b) K_5

Komponente

Definition

Ein maximale Teilgraph $G' = (V', G') \subset G$ mit $\forall v \in V' \forall w \in V' : v \overset{*}{\Rightarrow} w$ heißt Komponente von G

Separator

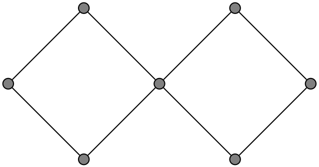
Definition

Gegeben ein Graph $G = (V, E)$ heißt ein Knoten $u \in V$ Separator, wenn für alle Pfade $p = v \overset{*}{\Rightarrow} w$ mit $v, w \in V$ gilt: $u \in p$

Separator

Definition

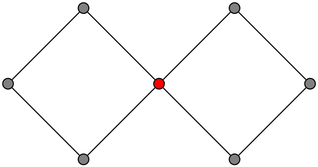
Gegeben ein Graph $G = (V, E)$ heißt ein Knoten $u \in V$ Separator, wenn für alle Pfade $p = v \overset{*}{\Rightarrow} w$ mit $v, w \in V$ gilt: $u \in p$



Separator

Definition

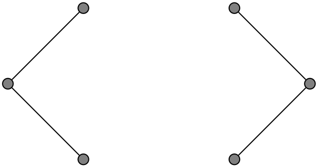
Gegeben ein Graph $G = (V, E)$ heißt ein Knoten $u \in V$ Separator, wenn für alle Pfade $p = v \overset{*}{\Rightarrow} w$ mit $v, w \in V$ gilt: $u \in p$



Separator

Definition

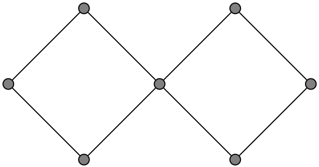
Gegeben ein Graph $G = (V, E)$ heißt ein Knoten $u \in V$ Separator, wenn für alle Pfade $p = v \overset{*}{\Rightarrow} w$ mit $v, w \in V$ gilt: $u \in p$



Bikomponente

Definition

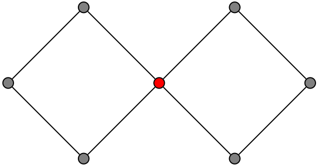
Ein maximale Teilgraph $G' = (V', G') \subset G$ ohne Separatoren heißt Bi-Komponente von G



Bikomponente

Definition

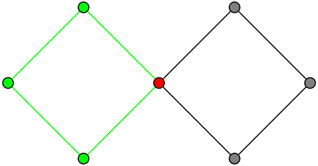
Ein maximale Teilgraph $G' = (V', G') \subset G$ ohne Separatoren heißt Bi-Komponente von G



Bikomponente

Definition

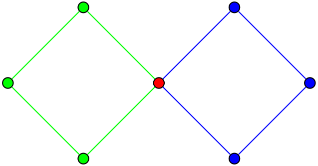
Ein maximale Teilgraph $G' = (V', G') \subset G$ ohne Separatoren heißt Bi-Komponente von G



Bikomponente

Definition

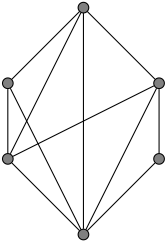
Ein maximale Teilgraph $G' = (V', G') \subset G$ ohne Separatoren heißt Bi-Komponente von G



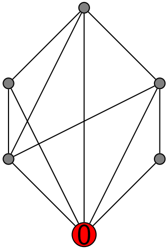
Der Algorithmus

1. Erstelle den Palmtree zu G

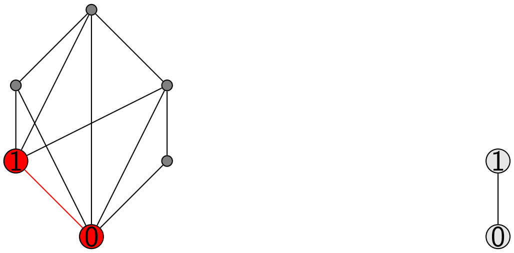
Palmtree



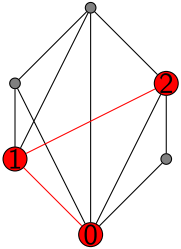
Palmtree



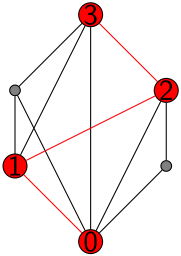
Palmtree

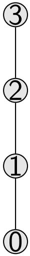


Palmtree

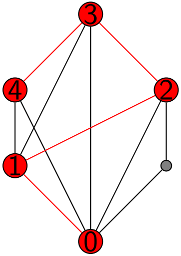


Palmtree

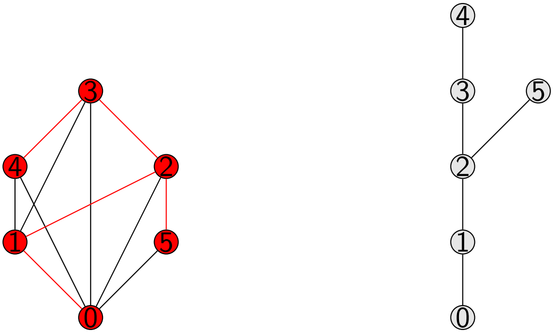




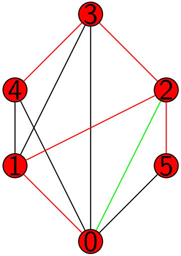
Palmtree

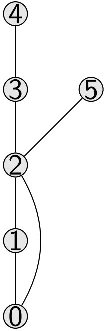


Palmtree

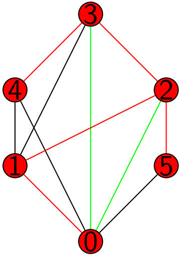


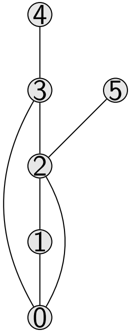
Palmtree



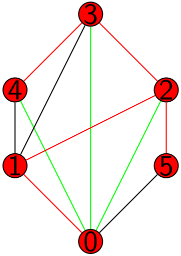


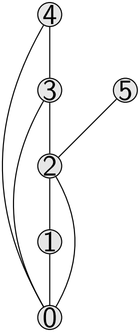
Palmtree



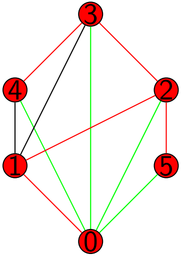


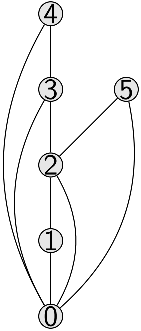
Palmtree



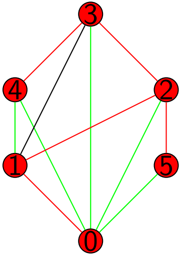


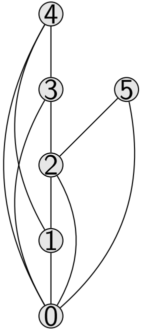
Palmtree



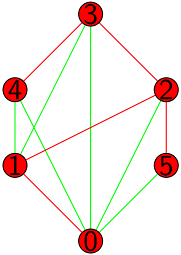


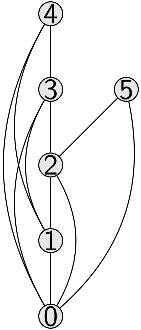
Palmtree





Palmtree

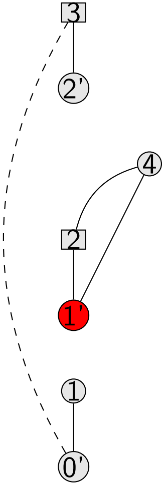




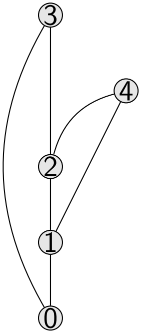
Der Algorithmus

1. Erstelle den Palmtree zu G
2. Berechne die Lowpoint-Werte der Knoten
3. Erstelle für alle Knoten v eine AdjacencyList, welche die Kinder von v sortiert nach Lowpoint enthält

Außen Aktive Knoten

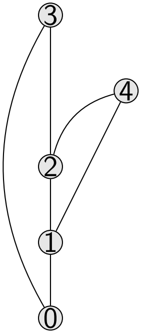


Lowpoint Werte



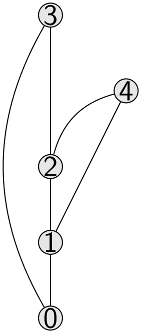
$L(4) = 2$

Lowpoint Werte



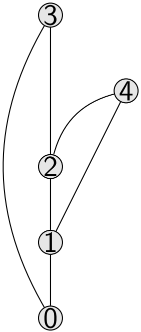
$L(4) = 2$
$L(3) = 0$

Lowpoint Werte



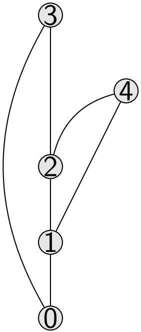
$$L(4) = 2$$
$$L(3) = 0$$
$$L(2) = 0$$

Lowpoint Werte



$L(4) = 2$
$L(3) = 0$
$L(2) = 0$
$L(1) = 0$

Lowpoint Werte



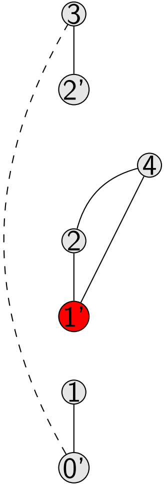
$$L(4) = 2$$
$$L(3) = 0$$
$$L(2) = 0$$
$$L(1) = 0$$
$$L(0) = 0$$

Außen Aktive Knoten per Lowpoint bestimmen

Lemma

Ein Knoten p heißt Außen Aktiv, wärend der Bearbeitung von w, genau dan wenn $L(p) < DFI(w)$ oder wenn $L(q) < DFI(w)$, wobei q das erste Element der AdjacencyList von p ist.

Außen Aktive Knoten per Lowpoint bestimmen



Lemma

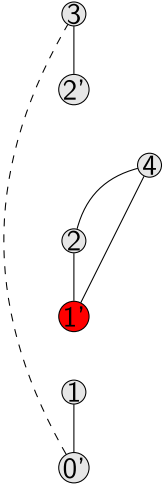
Ein Knoten p heißt Außen Aktiv, wärend der Bearbeitung von w, genau dan wenn $L(p) < DFI(w)$ oder wenn $L(q) < DFI(w)$, wobei q das erste Element der AdjacencyList von p ist.

Außen Aktive Knoten per Lowpoint bestimmen

<u>p = 2:</u>
$L(p) < 1 \Leftrightarrow 0 < 1$
$\Rightarrow$ 2 ist Außen Aktiv

Lemma

Ein Knoten p heißt Außen Aktiv, wärend der Bearbeitung von w, genau dan wenn $L(p) < DFI(w)$ oder wenn $L(q) < DFI(w)$, wobei q das erste Element der AdjacencyList von p ist.

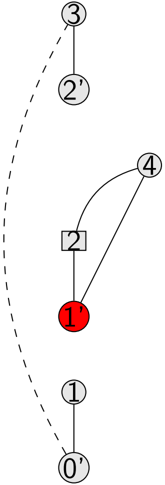


Außen Aktive Knoten per Lowpoint bestimmen

<u>p = 3:</u>
$L(p) < 1 \Leftrightarrow 0 < 1$
$\Rightarrow$ 3 ist Außen Aktiv

Lemma

Ein Knoten p heißt Außen Aktiv, wärend der Bearbeitung von w, genau dan wenn $L(p) < DFI(w)$ oder wenn $L(q) < DFI(w)$, wobei q das erste Element der AdjacencyList von p ist.

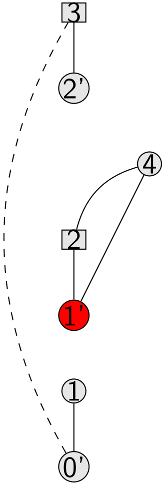


Außen Aktive Knoten per Lowpoint bestimmen

<u>p = 4:</u>
$L(p) < 1 \Leftrightarrow 2 < 1$
AdjacencyList von 4 ist leer
$\Rightarrow$ 4 ist nicht außen Aktiv

Lemma

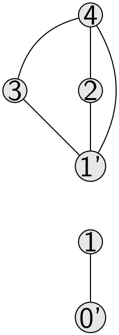
Ein Knoten p heißt Außen Aktiv, wärend der Bearbeitung von w, genau dan wenn $L(p) < DFI(w)$ oder wenn $L(q) < DFI(w)$, wobei q das erste Element der AdjacencyList von p ist.



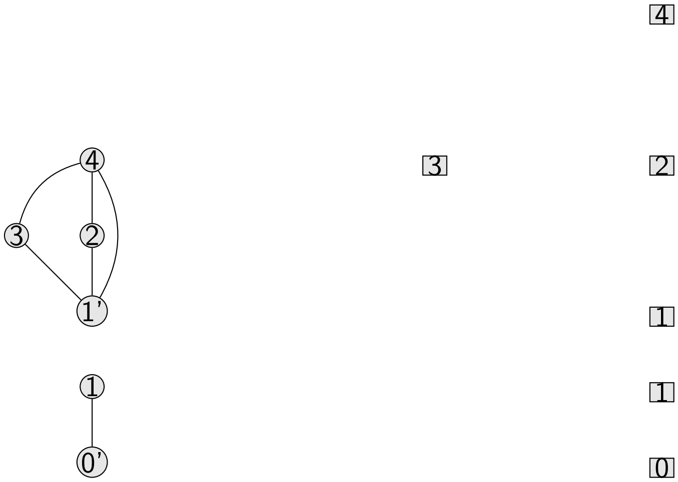
Der Algorithmus

1. Erstelle den Palmtree zu G
2. Berechne die Lowpoint-Werte der Knoten
3. Erstelle für alle Knoten v eine AdjacencyList, welche die Kinder von v sortiert nach Lowpoint enthält
4. Erstelle für alle Kanten e des DFS-Baums eine Bikomponente B und bette e in diese ein

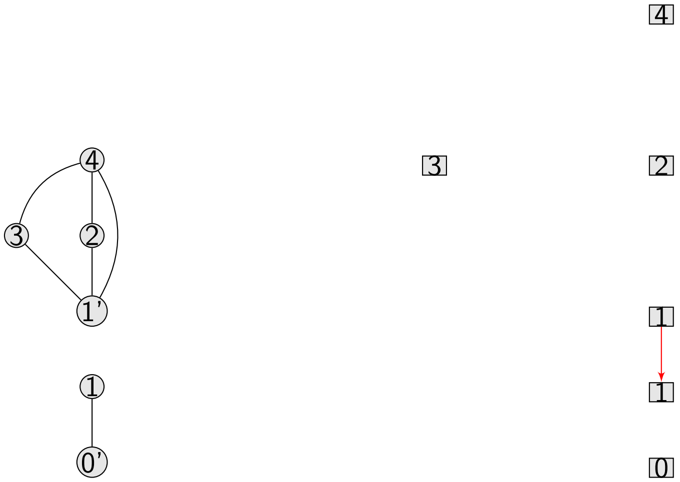
Die Datenstruktur



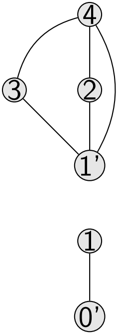
Die Datenstruktur

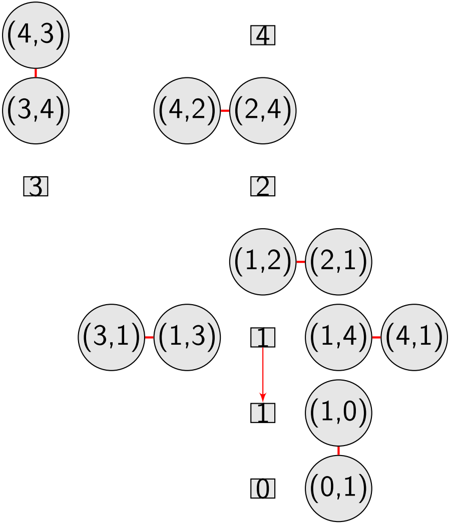


Die Datenstruktur

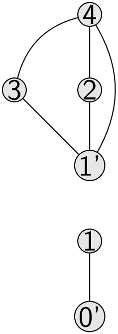


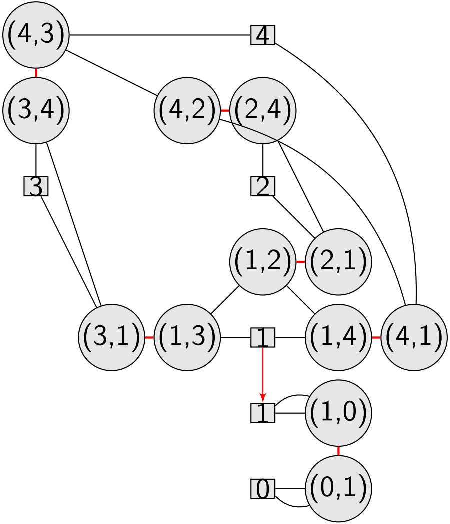
Die Datenstruktur



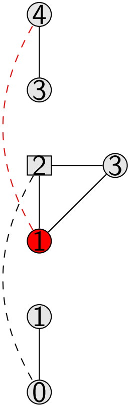


Die Datenstruktur

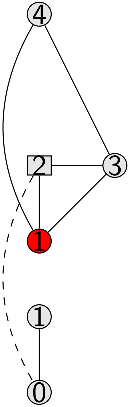




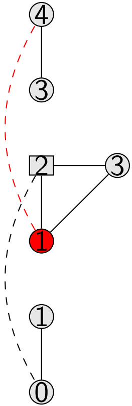
Zusammenfügen von Bikomponenten



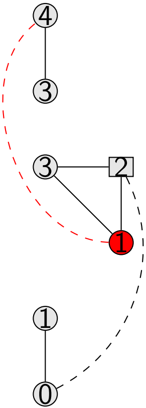
Zusammenfügen von Bikomponenten



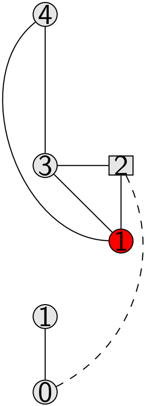
Zusammenfügen von Bikomponenten



Zusammenfügen von Bikomponenten



Zusammenfügen von Bikomponenten

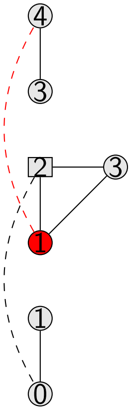


Der Algorithmus

1. Erstelle den Palmtree zu G
2. Berechne die Lowpoint-Werte der Knoten
3. Erstelle für alle Knoten v eine AdjacencyList, welche die Kinder von v sortiert nach Lowpoint enthält
4. Erstelle für alle Kanten e des DFS-Baums eine Bikomponente B und bette e in diese ein
5. Für alle Knoten v in inverser DFI-Ordnung:
6. Performe für alle Kanten, die von einem DFS Nachkommen von w nach v führen WalkUp

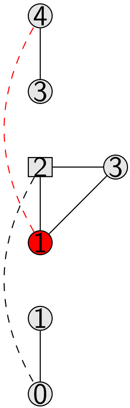
WalkUp

Suche einen Pfad von w nach v über die außen liegenden Knoten



WalkUp

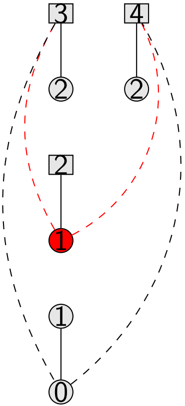
Suche einen Pfad von w nach v über die außen liegenden Knoten



Beachte:
Außenaktive Knoten dürfen nicht traversiert werden

WalkUp

Suche einen Pfad von w nach v über die außen liegenden Knoten

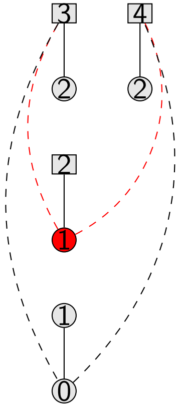


<u>Beachte:</u>

Wird eine Wurzel aus Zwei Außenaktiven Bikomponenten, oder aus zwei Richtungen beschritten, müssen zwei Pfade über die außen liegenden Knoten gefunden werden

WalkUp

Suche einen Pfad von w nach v über die außen liegenden Knoten



Beachte:

Eine Außen Aktive Wurzel, die nicht durch die Bikomp aus der man kommt außenaktiv ist, darf nur von einer Richtung aus beschritten werden

Relevante Bikomponente

Definition

Eine Bikomponente heißt relevent, wenn man wärend des WalkUps aus ihr herausgetreten ist

Der Algorithmus

1. Erstelle den Palmtree zu G
2. Berechne die Lowpoint-Werte der Knoten
3. Erstelle für alle Knoten v eine AdjacencyList, welche die Kinder von v sortiert nach Lowpoint enthält
4. Erstelle für alle Kanten e des DFS-Baums eine Bikomponente B und bette e in diese ein
5. Für alle Knoten v in inverser DFI-Ordnung:
6. Performe für alle Kanten, die von einem DFS Nachkommen von w nach v führen WalkUp
7. Performe für die gefundenen Pfade WalkDown

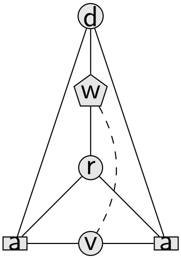
Walk Down

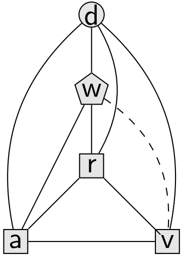
Performe DFS auf den Graphen, der durch die Pfade aus dem WalkUp induziert wird

Beachte jedoch

1. Füge zuerst die Kante zu einbettung hinzu, die an dem Knoten anliegt, indem du gerade bist
2. Gibt es eine relevante Bikomponte,ohne außen aktive Knoten von der der Momentane Knoten die Wurzel ist, schreite in diese Bikomponente hinein
3. Gibt es eine relevante Bikomponte,mit außen aktive Knoten von der der Momentane Knoten die Wurzel ist, schreite in diese Bikomponente hinein

Kuratowski Minore - Walk Up





Vielen Dank für ihre Aufmersamkeit