Karlsruher Institut für Technologie

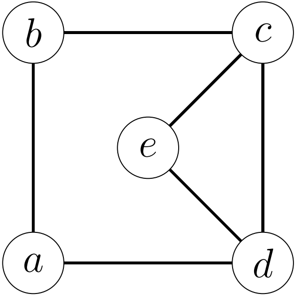
Algorithmen für Planare Graphen

21. Mai 2019, Übung 3

Guido Brückner

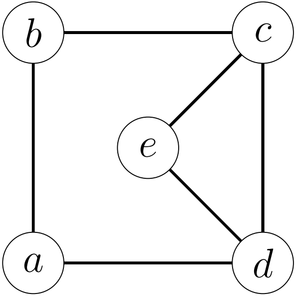
INSTITUT FÜR THEORETISCHE INFORMATIK

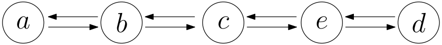
Erweiterte Inzidenzlisten



- Liste aller Knoten
- Knoten: Liste gerichteter Kanten entgegen dem Uhrzeigersinn
- Kante:
  - Zeiger auf entgegengerichtete Kante
  - Zeiger auf Ursprungsknoten

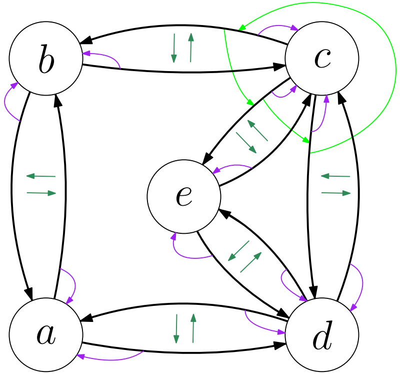
Erweiterte Inzidenzlisten

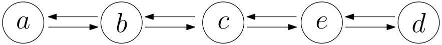




- Liste aller Knoten
- Knoten: Liste gerichteter Kanten entgegen dem Uhrzeigersinn
- Kante:
  - Zeiger auf entgegengerichtete Kante
  - Zeiger auf Ursprungsknoten

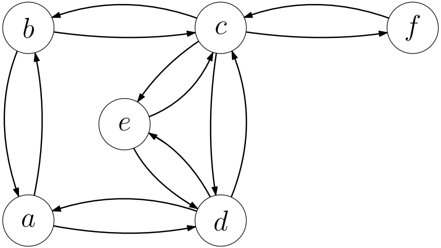
Erweiterte Inzidenzlisten





- Liste aller Knoten
- Knoten: Liste gerichteter Kanten entgegen dem Uhrzeigersinn
- Kante:
  - Zeiger auf entgegengerichtete Kante
  - Zeiger auf Ursprungsknoten

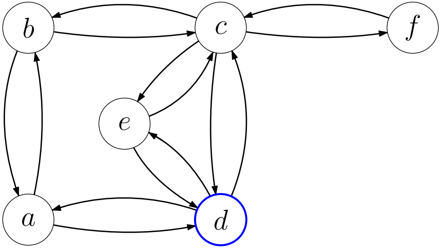
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

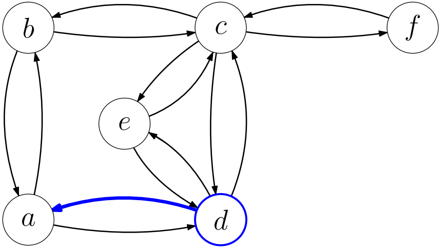
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

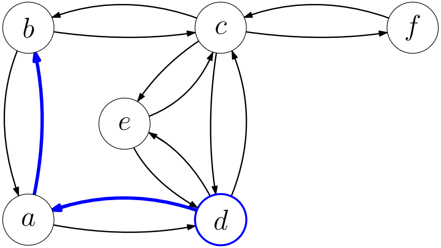
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

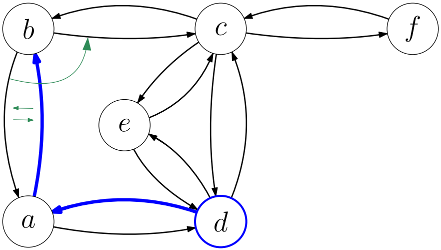
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

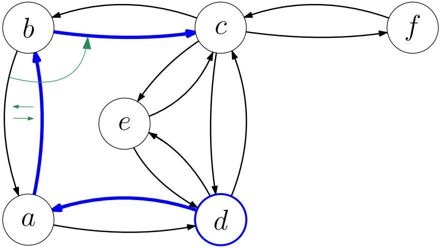
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

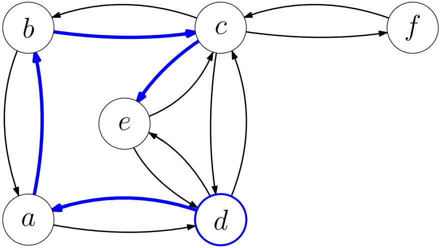
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

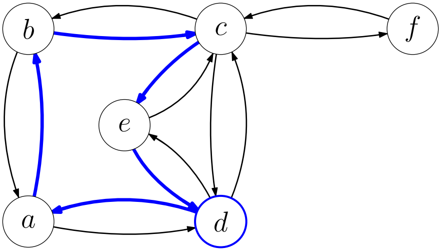
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.

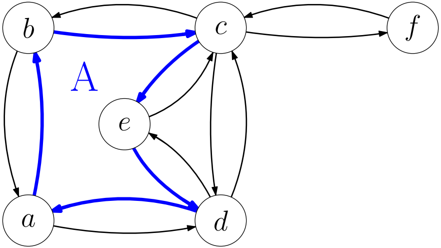
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

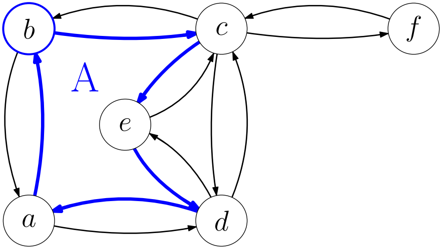
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

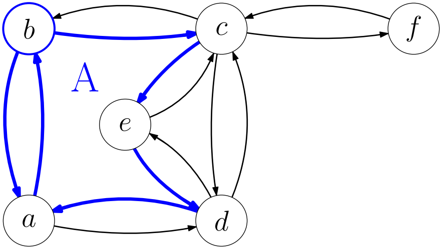
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

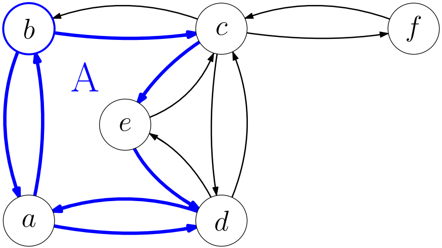
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

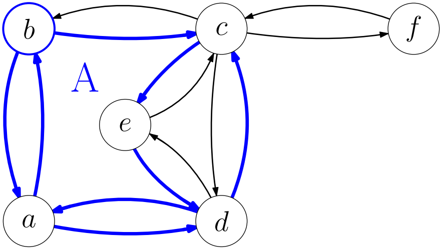
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

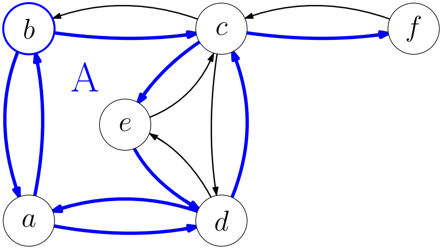
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

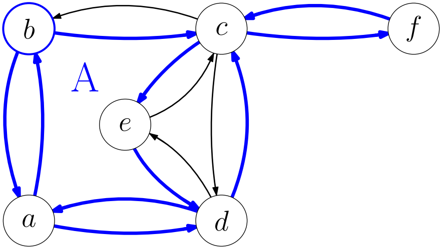
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

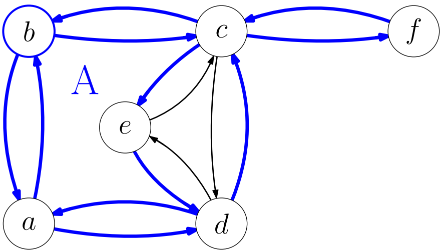
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

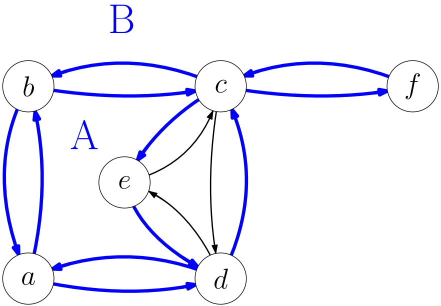
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

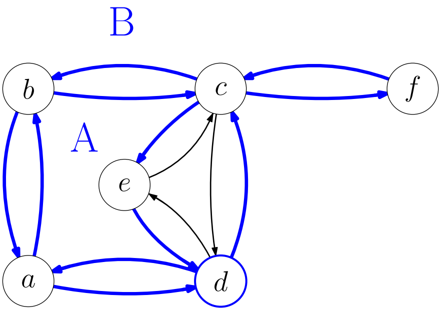
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

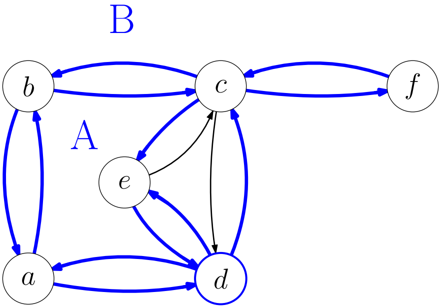
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

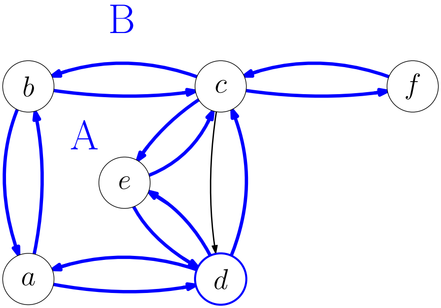
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

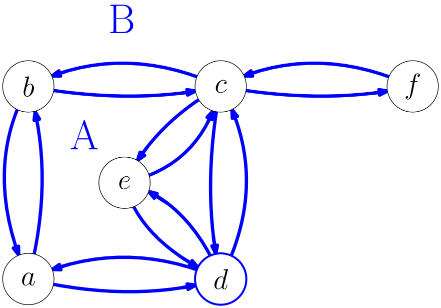
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

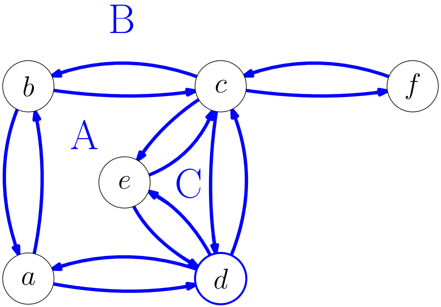
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

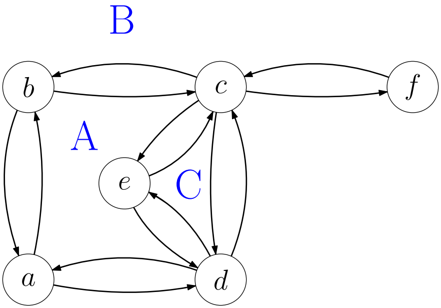
2 – Dualgraph



Duale Knoten

- Wähle einen Konten und eine ausgehende Kante beliebig.
- Beim nächsten Knoten wähle erste Kante gegen den Uhrzeigersinn.
- Markiere jede traversierte Kante mit der aktuellen *Facetten-ID*.
- Erhöhe die *Facetten-ID* wenn wieder beim Ausgangsknoten angekommen.

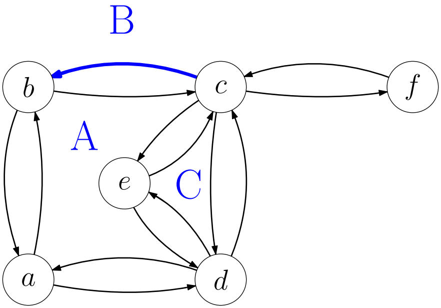
2 – Dualgraph



Duale Kanten

- Traversiere erneut jede Facette.
- Betrachte Hin- und Rückkante und füge für die dualen Knoten mit den entsprechenden *Facetten-IDs* eine duale Kante ein.
- Die dualen Kanten werden in der richtigen Reihenfolge eingefügt.

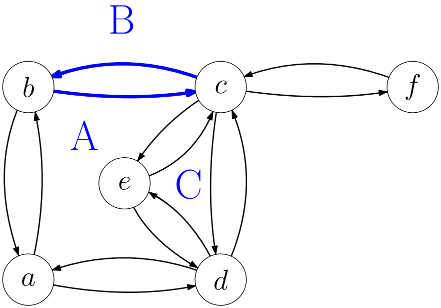
2 – Dualgraph



Duale Kanten

- Traversiere erneut jede Facette.
- Betrachte Hin- und Rückkante und füge für die dualen Knoten mit den entsprechenden *Facetten-IDs* eine duale Kante ein.
- Die dualen Kanten werden in der richtigen Reihenfolge eingefügt.

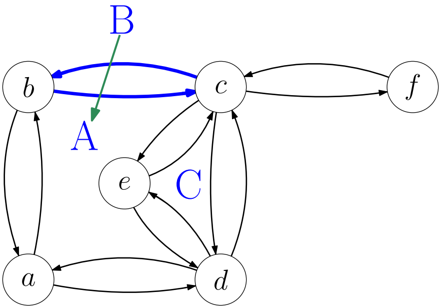
2 – Dualgraph



Duale Kanten

- Traversiere erneut jede Facette.
- Betrachte Hin- und Rückkante und füge für die dualen Knoten mit den entsprechenden *Facetten-IDs* eine duale Kante ein.
- Die dualen Kanten werden in der richtigen Reihenfolge eingefügt.

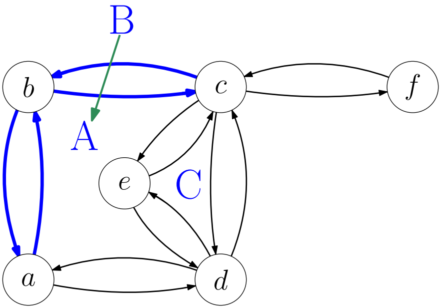
2 – Dualgraph



Duale Kanten

- Traversiere erneut jede Facette.
- Betrachte Hin- und Rückkante und füge für die dualen Knoten mit den entsprechenden *Facetten-IDs* eine duale Kante ein.
- Die dualen Kanten werden in der richtigen Reihenfolge eingefügt.

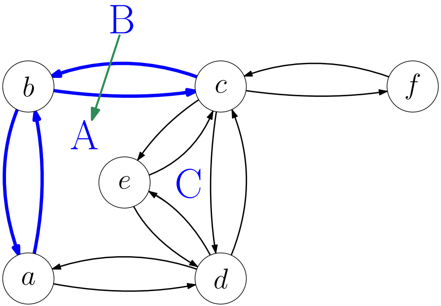
2 – Dualgraph



Duale Kanten

- Traversiere erneut jede Facette.
- Betrachte Hin- und Rückkante und füge für die dualen Knoten mit den entsprechenden *Facetten-IDs* eine duale Kante ein.
- Die dualen Kanten werden in der richtigen Reihenfolge eingefügt.

2 – Dualgraph



Rückkanten

- Speichere beim Einfügen der Kanten Zeiger von Kante im Ausgangsgraphen auf die Kante im Dualgraphen
- Wenn beim Einfügen einer Kante die Rückkante im Ausgangsgraphen schon einen Zeiger hat, ergänze beide Zeiger im Dualgraphen

1 – Triangulierung

Triangulierung in $\mathcal{O}(n)$

1. Füge Kanten hinzu so, dass es keine Grad-1 Knoten mehr gibt.
2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.
3. Löse Schleifen und Multikanten durch Kantentausch auf.

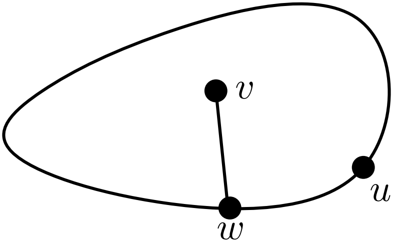
1 – Triangulierung

Triangulierung in $\mathcal{O}(n)$

1. Füge Kanten hinzu so, dass es keine Grad-1 Knoten mehr gibt.
2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.
3. Löse Schleifen und Multikanten durch Kantentausch auf.

1.1 – Triangulierung

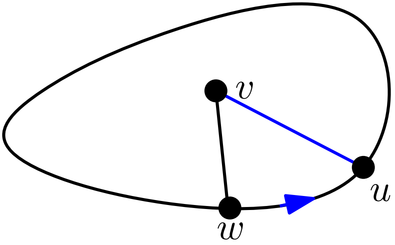
1. Füge Kanten hinzu so, dass es keine Grad-1-Knoten mehr gibt.



- Sei *v* Knoten mit Grad 1.
- Seine Kante sei $\{v, w\}$ und *f* die Facette in der er liegt.
- Laufe *f* von *w* aus im Gegenuhrzeigersinn ab.
- Verbinde *v* mit dem zweiten besuchten Knoten *u*.

1.1 – Triangulierung

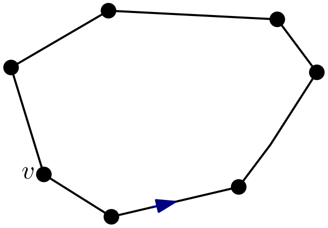
1. Füge Kanten hinzu so, dass es keine Grad-1-Knoten mehr gibt.



- Sei v Knoten mit Grad 1.
- Seine Kante sei $\{v, w\}$ und f die Facette in der er liegt.
- Laufe f von w aus im Gegenuhrzeigersinn ab.
- Verbinde v mit dem zweiten besuchten Knoten u.

1.2 – Triangulierung

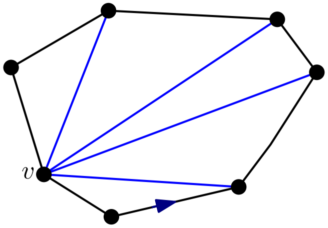
2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.



- Für jede Facette f wähle beliebigen Knoten v.
- Laufe f ab und verbinde v mit allen besuchten Knoten, außer dem Vorgänger und Nachfolger von v auf f.

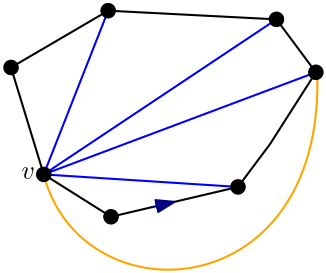
1.2 – Triangulierung

2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.



- Für jede Facette f wähle beliebigen Knoten v.
- Laufe f ab und verbinde v mit allen besuchten Knoten, außer dem Vorgänger und Nachfolger von v auf f.

2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.

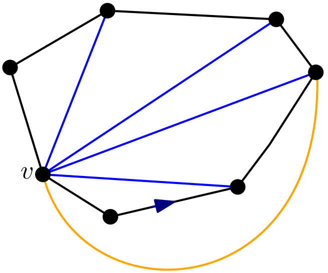


- Für jede Facette f wähle beliebigen Knoten v.
- Laufe f ab und verbinde v mit allen besuchten Knoten, außer dem Vorgänger und Nachfolger von v auf f.

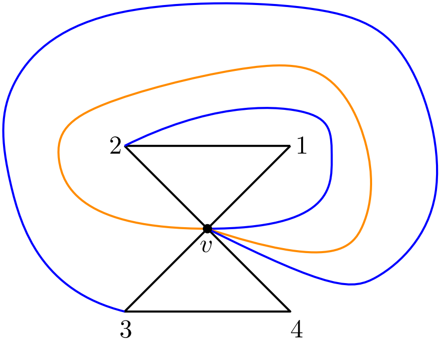
Doppelkanten

1.2 – Triangulierung

2. Trianguliere Graph ohne auf Schleifen/Multikanten zu achten.



Doppelkanten



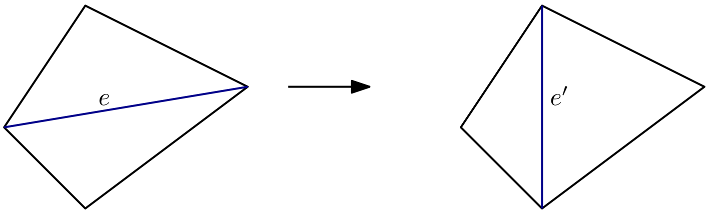
Schleifen

1.3 – Triangulierung

3. Löse Schleifen und Multikanten durch Kantentausch auf.

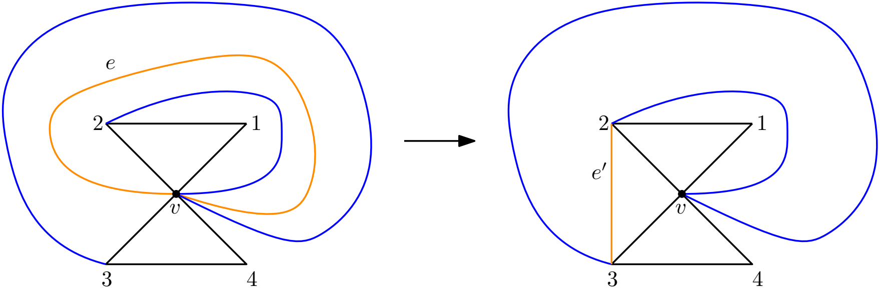
Kantentausch

- Betrachte Kante e eines triangulierten Graphen.
- Das Entfernen von e ergibt eine Facette f mit Grad 4.
- Füge Kante e' in f ein, die nicht die gleichen Knoten wie e verbindet.



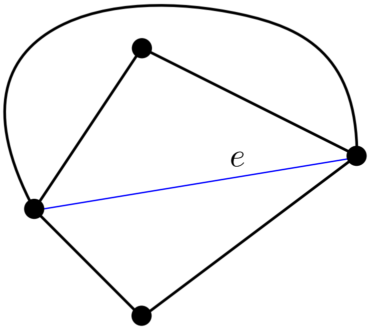
1.3 – Triangulierung

3. Löse **Schleifen** und Multikanten durch Kantentausch auf.



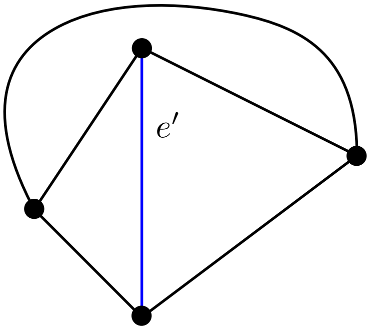
1.3 – Triangulierung

3. Löse Schleifen und **Multikanten** durch Kantentausch auf.
   - Sei *e* eine Multikanten.
   - Kann e' eine Multikante sein?



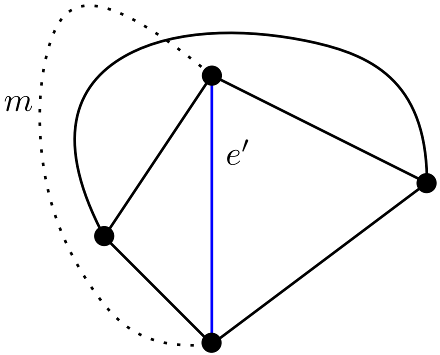
1.3 – Triangulierung

3. Löse Schleifen und **Multikanten** durch Kantentausch auf.
   - Sei e eine Multikanten.
   - Kann e' eine Multikante sein?



1.3 – Triangulierung

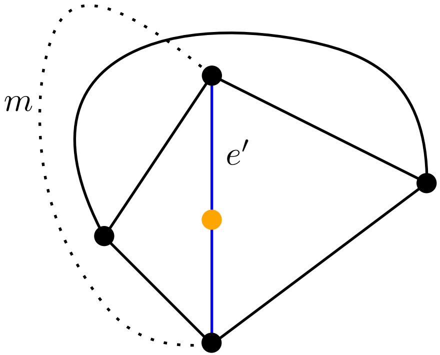
3. Löse Schleifen und **Multikanten** durch Kantentausch auf.



- Sei *e* eine Multikanten.
- Kann e' eine Multikante sein?
- Angenommen es gibt eine planare Einbettung mit Kante *m*.

1.3 – Triangulierung

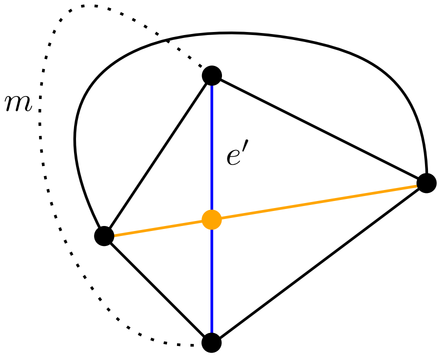
3. Löse Schleifen und **Multikanten** durch Kantentausch auf.



- Sei *e* eine Multikanten.
- Kann e' eine Multikante sein?
- Angenommen es gibt eine planare Einbettung mit Kante *m*.
- Unterteile e'.

1.3 – Triangulierung

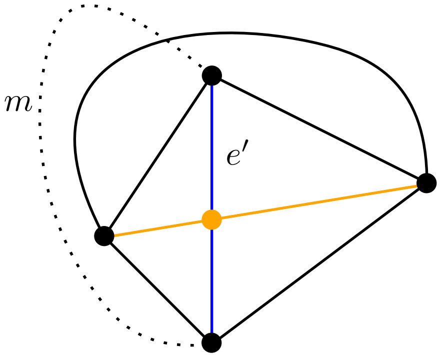
3. Löse Schleifen und **Multikanten** durch Kantentausch auf.



- Sei e eine Multikanten.
- Kann e' eine Multikante sein?
- Angenommen es gibt eine planare Einbettung mit Kante m.
- Unterteile e'.
- Füge weitere Kanten ein, die die Planarität nicht verletzen.

1.3 – Triangulierung

3. Löse Schleifen und **Multikanten** durch Kantentausch auf.



- Sei *e* eine Multikanten.
- Kann e' eine Multikante sein?
- Angenommen es gibt eine planare Einbettung mit Kante *m*.
- Unterteile e'.
- Füge weitere Kanten ein, die die Planarität nicht verletzen.
- Planare Einbettung für K_5 gefunden. $\lightning$

1 – Triangulierung Laufzeit

Wenn Schleifen und Multikanten bekannt sind, dann braucht Schritt 3 lineare Zeit: $\mathcal{O}(1)$ pro Facette.

Lemma

Doppelkanten und Schleifen können in $\mathcal{O}(n)$ Zeit bestimmt werden.

Für jeden Konten $v \in V$:

- Iteriere über alle ausgehende Kante von v und markiere benachbarte Knoten.
- Wird ein Knoten mehr als einmal markiert ist eine Multikante gefunden.
- Wird v markiert ist eine Schleife gefunden.

Jede *gerichtete* Kante wird einmal besucht $\Rightarrow \mathcal{O}(n)$

3 – Geradlinige Zeichnungen

Sei G ein einfacher planarer Graph, der kombinatorisch eingebettet ist. Zeigen Sie, dass G eine planare Zeichnung besitzt in der jede Kante durch eine Strecke repräsentiert wird.

- Wir führen den Beweis für alle zusammenhängenden maximal planaren Graphen.
- Sei G maximal planar.
- $\Rightarrow$ Jede Facette von G ist ein Dreieck.
- Seien u, v, w die Knoten der äußeren Facette.

Induktion über n:

- **IA:** $n = 3$
- **IV:** Jeder einfache, eingebettete, maximal planare Graph mit $n - 1$ Knoten lässt sich geradlinig zeichnen.

3 – Geradlinige Zeichnungen

Sei G ein einfacher planarer Graph, der kombinatorisch eingebettet ist. Zeigen Sie, dass G eine planare Zeichnung besitzt in der jede Kante durch eine Strecke repräsentiert wird.

- Wir führen den Beweis für alle zusammenhängenden maximal planaren Graphen.
- Sei G maximal planar.
- $\Rightarrow$ Jede Facette von G ist ein Dreieck.
- Seien u, v, w die Knoten der äußeren Facette.

Induktion über n:

- **IA:** $n = 3$
- **IV:** Jeder einfache, eingebettete, maximal planare Graph mit $n - 1$ Knoten lässt sich geradlinig zeichnen.

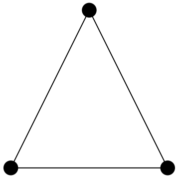
3 – Geradlinige Zeichnungen

Sei G ein einfacher planarer Graph, der kombinatorisch eingebettet ist. Zeigen Sie, dass G eine planare Zeichnung besitzt in der jede Kante durch eine Strecke repräsentiert wird.

- Wir führen den Beweis für alle zusammenhängenden maximal planaren Graphen.
- Sei G maximal planar.
- $\Rightarrow$ Jede Facette von G ist ein Dreieck.
- Seien u, v, w die Knoten der äußeren Facette.

Induktion über n:

- **IA:** $n = 3$
- **IV:** Jeder einfache, eingebettete, maximal planare Graph mit $n - 1$ Knoten lässt sich geradlinig zeichnen.



3.1 – Geradlinige Zeichnungen

$n-1 \curvearrowright n$

- Sei G ein maximal planarer Graph mit $n \geq 4$ Knoten.

Lemma

Es gibt mindestens 4 Knoten mit $\deg(v) \leq 5$.

- G maximal planar $\Rightarrow m = 3n - 6$
- Sei Defizit $def(v) := 6 - deg(v)$

Behauptung: $\Sigma_{v \in V} def(v) = 12$ Beweis: Tafel.

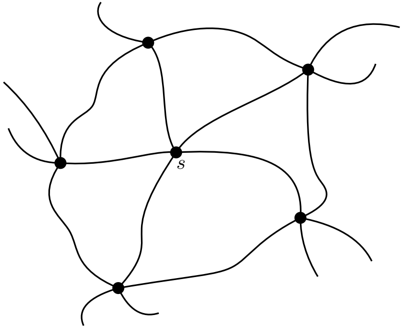
Behauptung: Jeder Knoten hat maximal Defizit 3. Beweis: Tafel.

$\Rightarrow$ Es gibt mindestens 4 Knoten mit $def(v) > 0$.
$\Rightarrow$ Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.

3.2 – Geradlinige Zeichnungen

$n-1 \curvearrowright n$

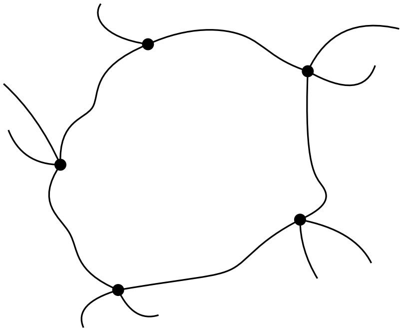
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



3.2 – Geradlinige Zeichnungen

$n - 1 \curvearrowright n$

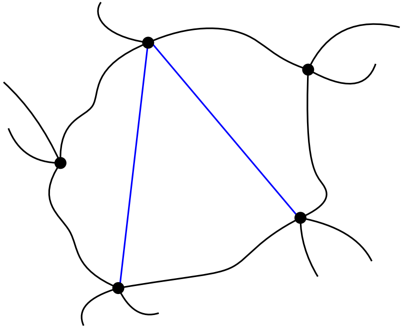
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



3.2 – Geradlinige Zeichnungen

$n-1 \curvearrowright n$

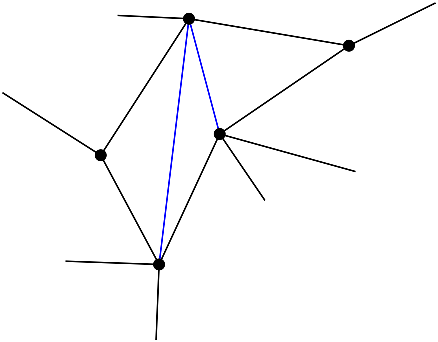
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



3.2 – Geradlinige Zeichnungen

$n - 1 \curvearrowright n$

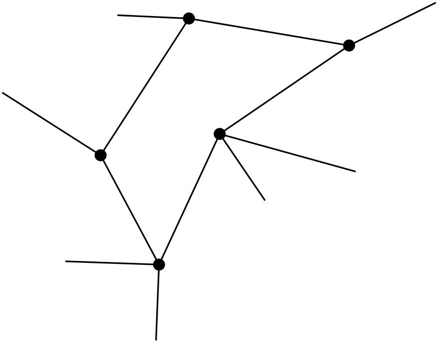
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



3.2 – Geradlinige Zeichnungen

$n-1 \curvearrowright n$

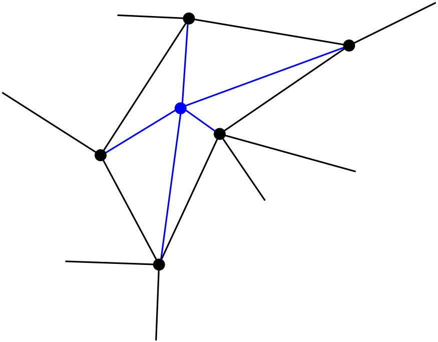
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



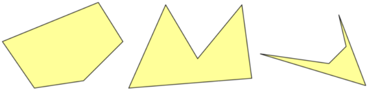
3.2 – Geradlinige Zeichnungen

$n-1 \curvearrowright n$

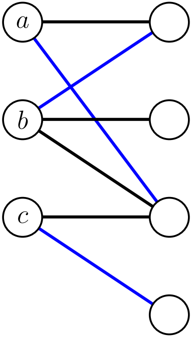
- Es gibt mindestens 4 Knoten mit $deg(v) \leq 5$.
- Wähle Knoten s mit $deg(s) \leq 5$ und $s \notin \{u, v, w\}$ (die Knoten der äußeren Facette).



- *Problem der Museumswächter*
- Zur Bewachung eines überschneidungsfreien, geschlossenen, planaren Polygons mit n Ecken sind maximal $\lfloor \frac{n}{3} \rfloor$ Wächter nötig.

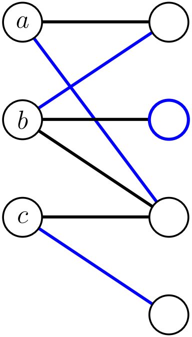


4 – Erhöhende Wege



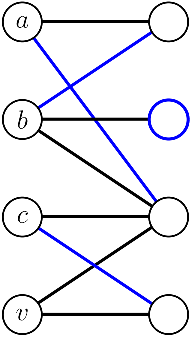
- Bestimme erhöhenden Weg bzgl. M' mit Endknoten v.
- Algorithmus soll in $\mathcal{O}(m)$ liegen.
- *Hinweis:* Modifizieren Sie eine Breitensuche mit Startknoten v.

4 – Erhöhende Wege



- Markiere ungematchte Knoten in $G - v$

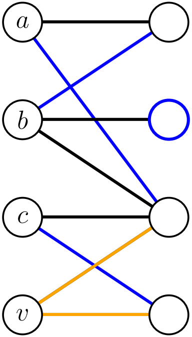
4 – Erhöhende Wege



v

- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.

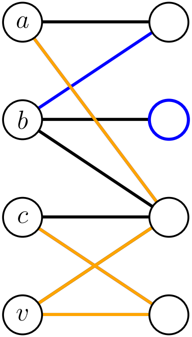
4 – Erhöhende Wege



v

- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.

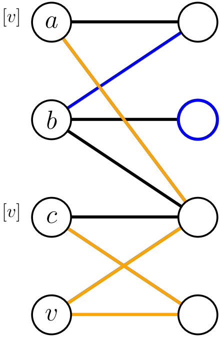
4 – Erhöhende Wege



v a c

- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.

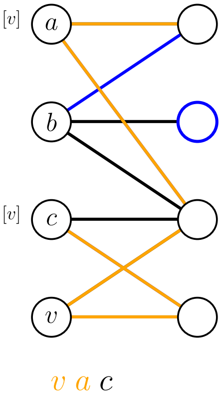
4 – Erhöhende Wege



v a c

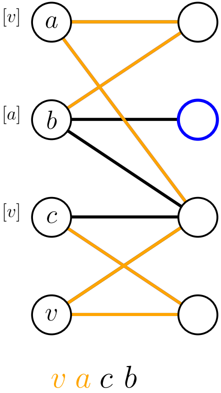
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.

4 – Erhöhende Wege



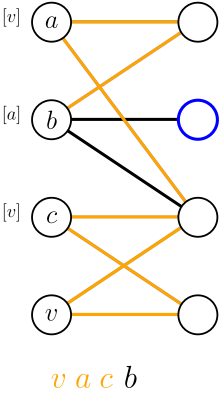
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.

4 – Erhöhende Wege



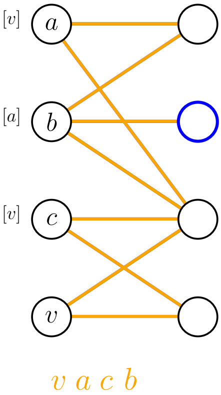
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.

4 – Erhöhende Wege



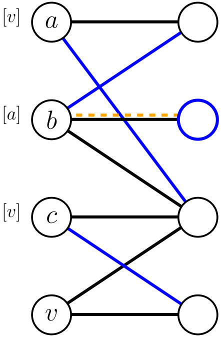
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.

4 – Erhöhende Wege



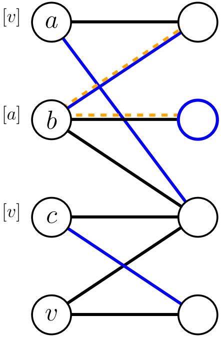
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.

4 – Erhöhende Wege



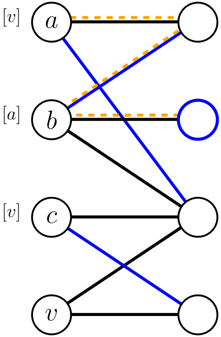
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

4 – Erhöhende Wege



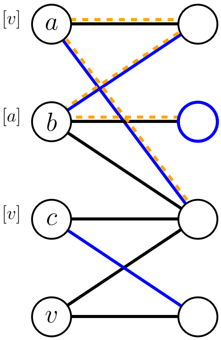
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

4 – Erhöhende Wege



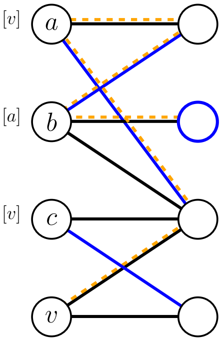
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

4 – Erhöhende Wege



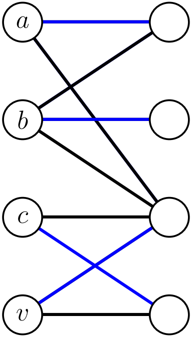
- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

4 – Erhöhende Wege



- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

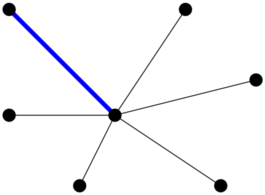
4 – Erhöhende Wege



- Markiere ungematchte Knoten in $G - v$
- BFS beginnend bei v.
- Füge Matchingknoten zur Queue hinzu.
- Brich ab, wenn markierter Knoten gefunden wird.
- Rekonstruiere erhöhenden Pfad.

5 – Große und kleine Matchings

Geben Sie für jedes $n \geq 2$ einen zusammenhängenden Graphen an, mit kardinalitätsmaximalem Matching der Größe: 1



5 – Große und kleine Matchings

Geben Sie für jedes $n \geq 2$ einen zusammenhängenden Graphen an, mit kardinalitätsmaximalem Matching der Größe: 1 und $\lfloor \frac{n}{2} \rfloor$