Karlsruher Institut für Technologie

Algorithmen II
Abschlussveranstaltung am 07.02.2013

Zusammenfassung & Anmerkungen zur Klausur

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Klausur

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet entweder direkt eingetragen oder bekommt einen blauen Schein, den ihr in unserem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreiben (unterschrieben im Sekretariat abgeben).

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet entweder direkt eingetragen oder bekommt einen blauen Schein, den ihr in unserem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreiben (unterschrieben im Sekretariat abgeben).

- Nicht mehr abmelden. (Eine Abmeldung ist noch unmittelbar vor Beginn der Klausur möglich.)

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet entweder direkt eingetragen oder bekommt einen blauen Schein, den ihr in unserem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreiben (unterschrieben im Sekretariat abgeben).

- Nicht mehr abmelden. (Eine Abmeldung ist noch unmittelbar vor Beginn der Klausur möglich.)
- Ab 26. Februar auf der Vorlesungshomepage nachschauen, welchem Hörsaal ihr zugeteilt seid.

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet entweder direkt eingetragen oder bekommt einen blauen Schein, den ihr in unserem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreiben (unterschrieben im Sekretariat abgeben).

- Nicht mehr abmelden. (Eine Abmeldung ist noch unmittelbar vor Beginn der Klausur möglich.)
- Ab 26. Februar auf der Vorlesungshomepage nachschauen, welchem Hörsaal ihr zugeteilt seid.
- Am 1. März um 10:45 Uhr bei dem entsprechenden Hörsaal erscheinen. Um 11:00 Uhr startet die 2-Stündige Klausur.

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet ent… direkt eingetragen oder bekommt einen blauen Schein, den ihr in …erem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreib… (unterschrieben im Sekretariat abgeben).

- Nicht mehr abmelden. (Eine Abmeldung ist noch unmitte…or Beginn der Klausur möglich.)
- Ab 26. Februar auf der Vorlesungshomepage nachscha…n, welchem Hörsaal ihr zugeteilt seid.
- Am 1. März um 10:45 Uhr bei dem entsprechenden Hörsaal erscheinen. Um 11:00 Uhr startet die 2-Stündige Klausur.

lernen nicht vergessen!

Organisatorisches

Was muss ich tun um an der Hauptklausur teilzunehmen?

- Bis zum **22. Februar** anmelden. (Spätere Anmeldung nicht möglich!)

  Normalfall: Anmeldung über das Studierendenportal

  Informationswirte u. ä.: Anmeldung im Studienbüro: ihr werdet entweder direkt eingetragen oder bekommt einen blauen Schein, den ihr in eurem Sekretariat abgeben müsst.

  Schülerstudenten: Anmeldung mit einem formlosen Schreiben (unterschrieben im Sekretariat abgeben).

- Nicht mehr abmelden. (Eine Abmeldung ist noch unmittelbar vor Beginn der Klausur möglich.)
- Ab 26. Februar auf der Vorlesungshomepage nachschauen, welchem Hörsaal ihr zugeteilt seid.
- Am 1. März um 10:45 Uhr bei dem entsprechenden Hörsaal erscheinen. Um 11:00 Uhr startet die 2-Stündige Klausur.

Die **Nachklausur** findet am 2. Oktober um 14:00 Uhr statt.

lernen nicht vergessen!

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.
- **Nicht** die Themen, die nur in der Übung behandelt wurden. Genauer:
  - Ein Problem, dass in der Übung, nicht aber in der Vorlesung behandelt wurde, wird als unbekannt angenommen.
  - Das heißt nicht, dass das Problem nicht in der Klausur definiert werden kann um in der Vorlesung gelernte Techniken darauf zu übertraten.

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.
- **Nicht** die Themen, die nur in der Übung behandelt wurden. Genauer:
  - Ein Problem, dass in der Übung, nicht aber in der Vorlesung behandelt wurde, wird als unbekannt angenommen.
  - Das heißt nicht, dass das Problem nicht in der Klausur definiert werden kann um in der Vorlesung gelernte Techniken darauf zu übertraten.

Womit kann ich lernen?

(außer mit diesen Folien hier)

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.
- **Nicht** die Themen, die nur in der Übung behandelt wurden. Genauer:
  - Ein Problem, dass in der Übung, nicht aber in der Vorlesung behandelt wurde, wird als unbekannt angenommen.
  - Das heißt nicht, dass das Problem nicht in der Klausur definiert werden kann um in der Vorlesung gelernte Techniken darauf zu übertraten.

Womit kann ich lernen? (außer mit diesen Folien hier)

- Vorlesungsfolien & Literaturhinweise (auf der Homepage)

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.
- **Nicht** die Themen, die nur in der Übung behandelt wurden. Genauer:
  - Ein Problem, dass in der Übung, nicht aber in der Vorlesung behandelt wurde, wird als unbekannt angenommen.
  - Das heißt nicht, dass das Problem nicht in der Klausur definiert werden kann um in der Vorlesung gelernte Techniken darauf zu übertraten.

Womit kann ich lernen? (außer mit diesen Folien hier)

- Vorlesungsfolien & Literaturhinweise (auf der Homepage)
- Übungsblätter & Übungsfolien (auf der Homepage)

Inhaltliches

Was erwartet mich in der Klausur?

- Themen, die in der Vorlesung behandelt wurden.
  - Definierte Probleme.
  - Vorgestellte Algorithmen.
  - Anwendung der gelernten Techniken auf andere Problemstellungen.
- **Nicht** die Themen, die nur in der Übung behandelt wurden. Genauer:
  - Ein Problem, dass in der Übung, nicht aber in der Vorlesung behandelt wurde, wird als unbekannt angenommen.
  - Das heißt nicht, dass das Problem nicht in der Klausur definiert werden kann um in der Vorlesung gelernte Techniken darauf zu übertraten.

Womit kann ich lernen? (außer mit diesen Folien hier)

- Vorlesungsfolien & Literaturhinweise (auf der Homepage)
- Übungsblätter & Übungsfolien (auf der Homepage)
- Alte Algorithmentechnik Klausuren (auf der Homepage)

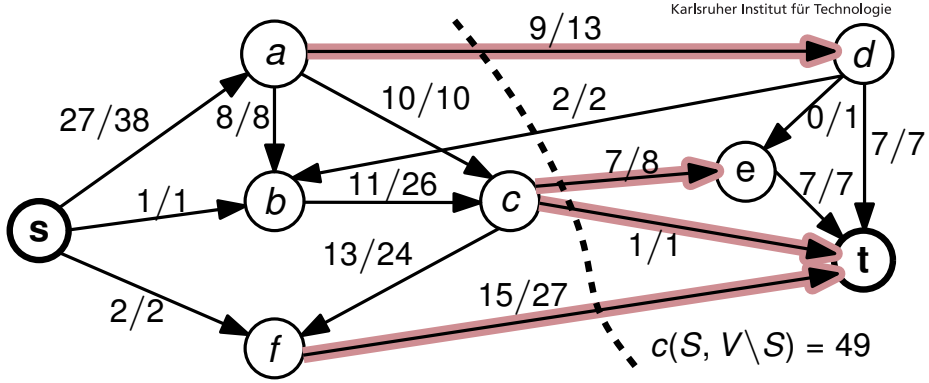
  Achtung: Die waren 1-stündig und der Stoff stimmt nicht komplett überein!

Zusammenfassung

(keine Gewährleistung für inhaltliche Vollständigkeit)

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

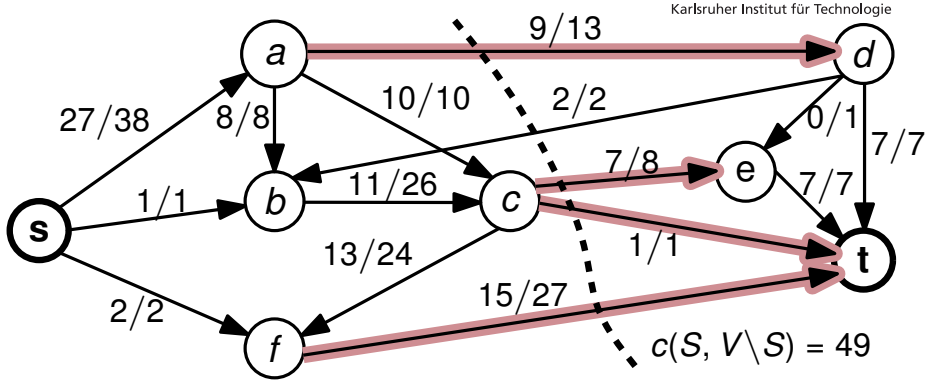
Flüsse & Schnitte



Grundlegendes:

- Was ist ein Flussnetzwerk?
- Welche Bedingungen stellt man an einen Fluss?
- Was ist ein minimaler Schnitt? Und was hat das mit Flüssen zu tun?
- Was ist das Residualnetzwerk und was hat es mit erhöhenden Wegen auf sich?

Flüsse & Schnitte



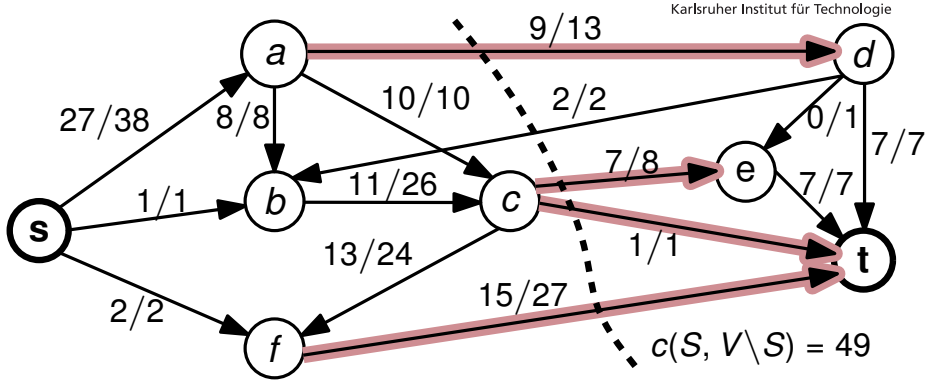
Grundlegendes:

- Was ist ein Flussnetzwerk?
- Welche Bedingungen stellt man an einen Fluss?
- Was ist ein minimaler Schnitt? Und was hat das mit Flüssen zu tun?
- Was ist das Residualnetzwerk und was hat es mit erhöhenden Wegen auf sich?

Algorithmen & Techniken:

- Ford-Fulkerson bzw. Edmonds-Karp:
  - Wie funktioniert's?
  - Was sind die Unterschiede?
  - Wie wirkt sich das auf die Laufzeit aus?

Flüsse & Schnitte



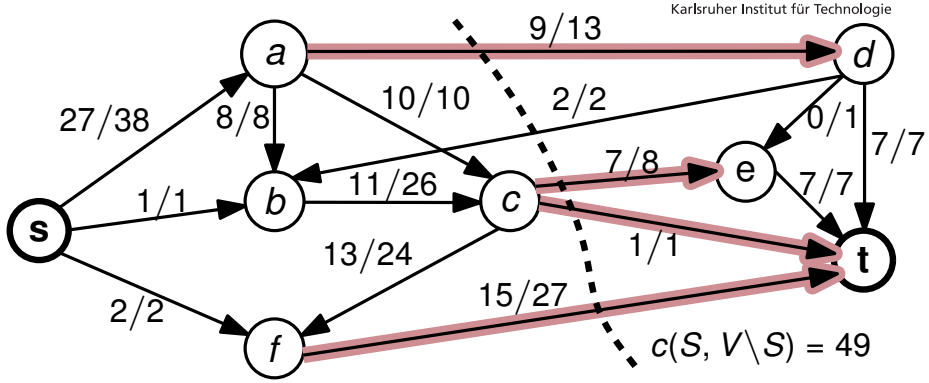
Grundlegendes:

- Was ist ein Flussnetzwerk?
- Welche Bedingungen stellt man an einen Fluss?
- Was ist ein minimaler Schnitt? Und was hat das mit Flüssen zu tun?
- Was ist das Residualnetzwerk und was hat es mit erhöhenden Wegen auf sich?

Algorithmen & Techniken:

- Ford-Fulkerson bzw. Edmonds-Karp:
  - Wie funktioniert's?
  - Was sind die Unterschiede?
  - Wie wirkt sich das auf die Laufzeit aus?
- Was ist ein lineares Programm (LP)? Wie kann man Flussprobleme mittels LP lösen? Warum muss man die Ganzzahligkeit nicht explizit fordern?

Flüsse & Schnitte



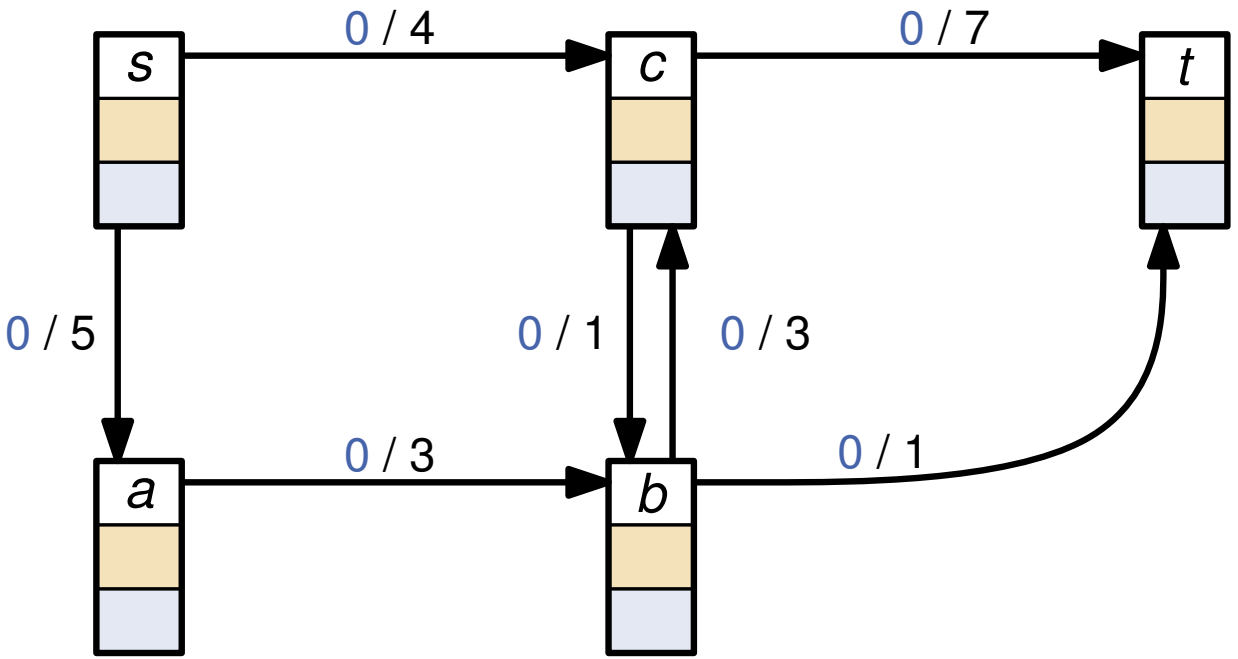
Grundlegendes:

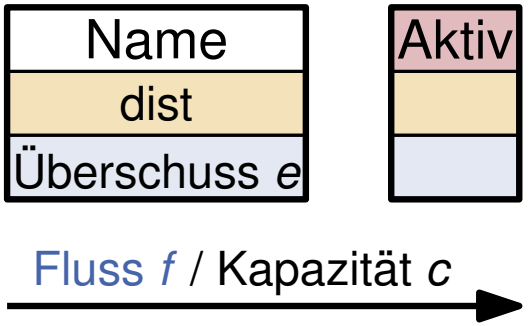
- Was ist ein Flussnetzwerk?
- Welche Bedingungen stellt man an einen Fluss?
- Was ist ein minimaler Schnitt? Und was hat das mit Flüssen zu tun?
- Was ist das Residualnetzwerk und was hat es mit erhöhenden Wegen auf sich?

Algorithmen & Techniken:

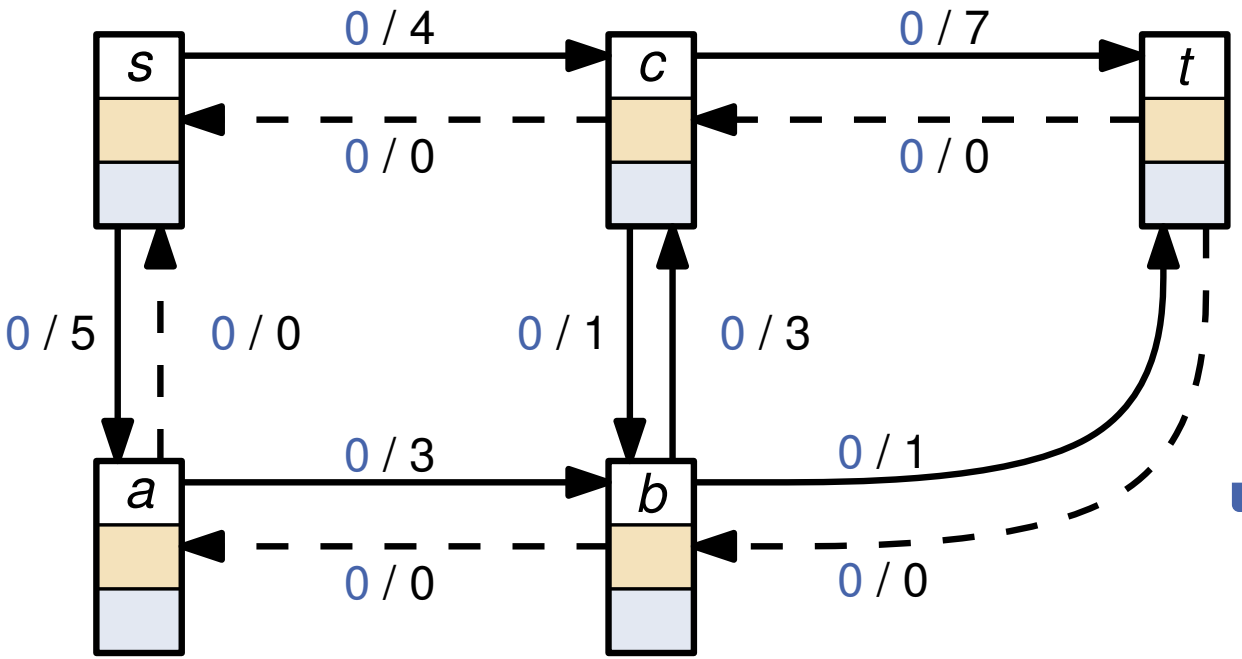
- Ford-Fulkerson bzw. Edmonds-Karp:
  - Wie funktioniert's?
  - Was sind die Unterschiede?
  - Wie wirkt sich das auf die Laufzeit aus?
- Was ist ein lineares Programm (LP)? Wie kann man Flussprobleme mittels LP lösen? Warum muss man die Ganzzahligkeit nicht explizit fordern?
- Algorithmus von Goldberg & Tarjan: siehe nächste Folie!

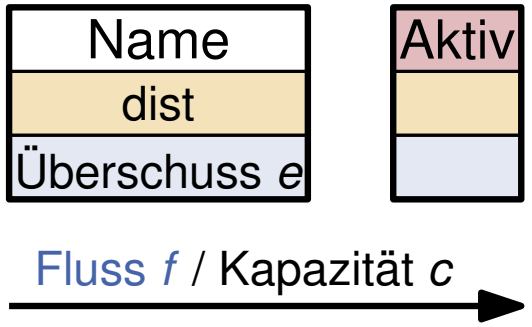
Goldberg & Tarjan





Goldberg & Tarjan

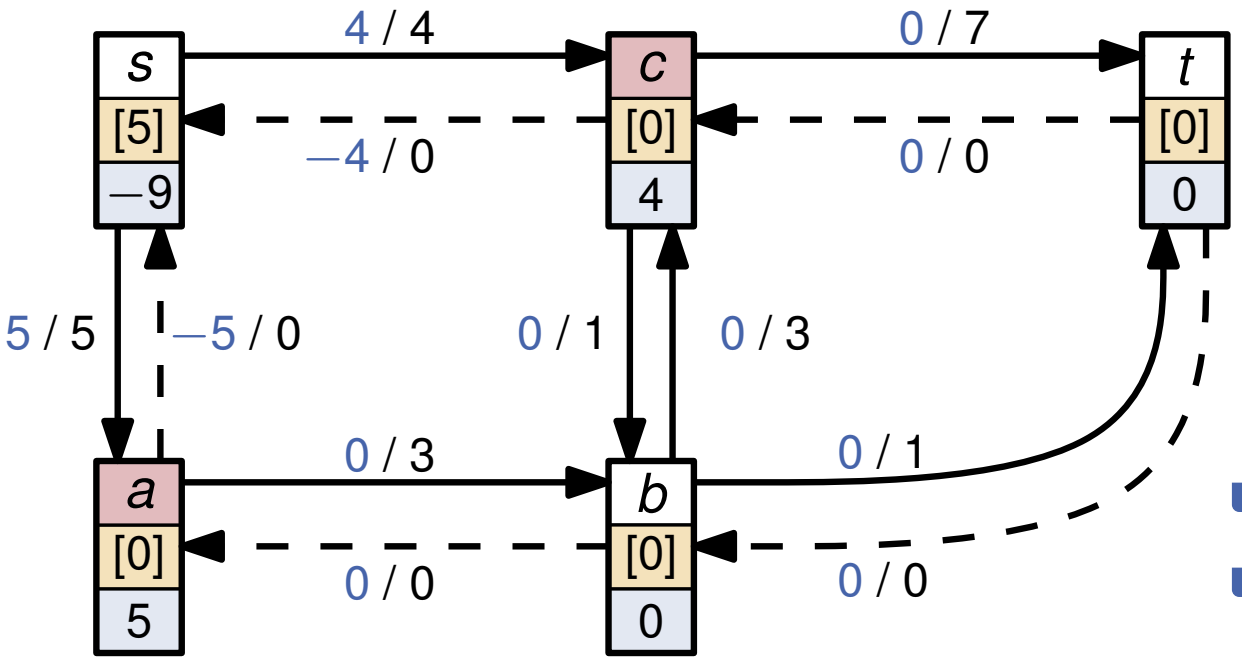


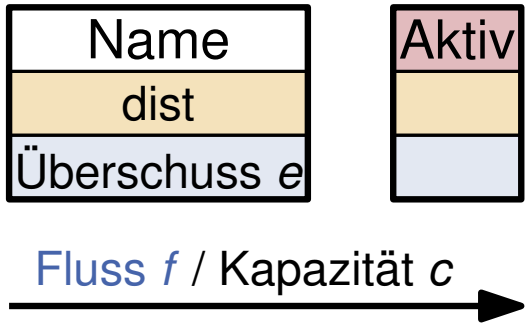


- Führe Gegenkanten ein

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

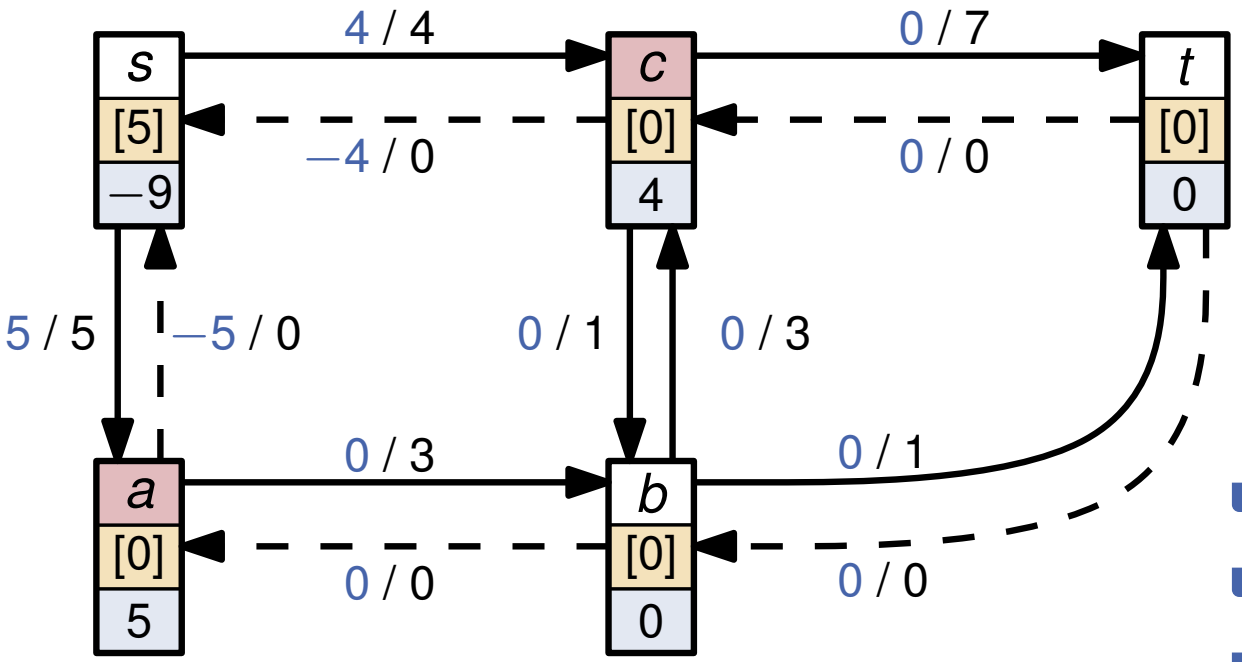
Goldberg & Tarjan

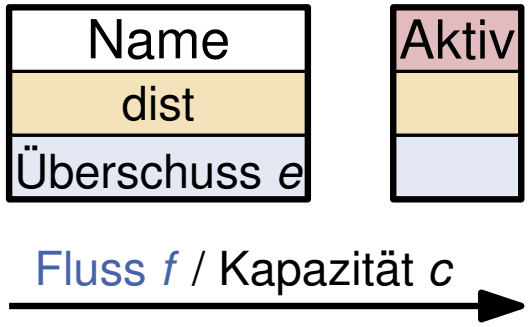




- Führe Gegenkanten ein
- Initialisiere dist und *f*

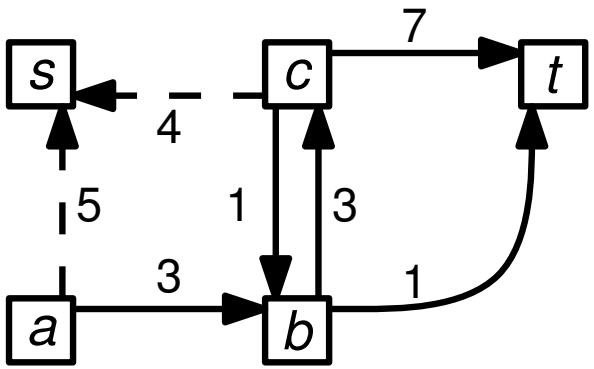
Goldberg & Tarjan



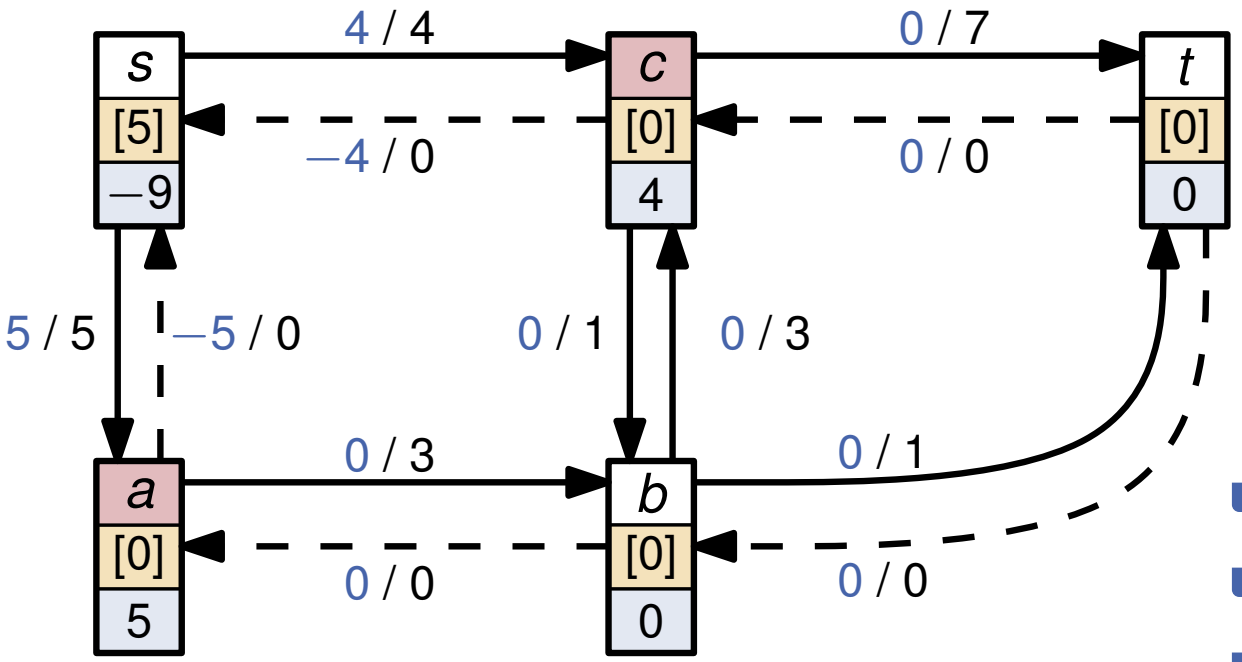


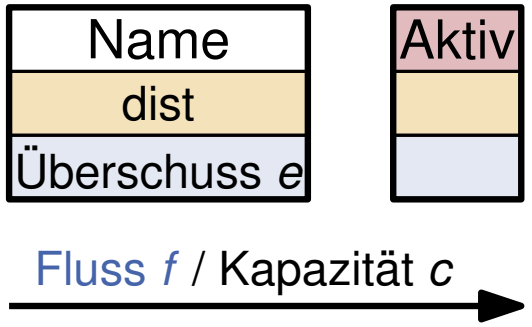
- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

Residualnetzwerk D_f:



Goldberg & Tarjan

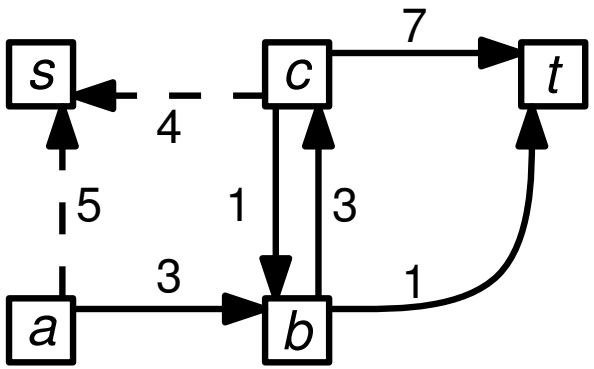




- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

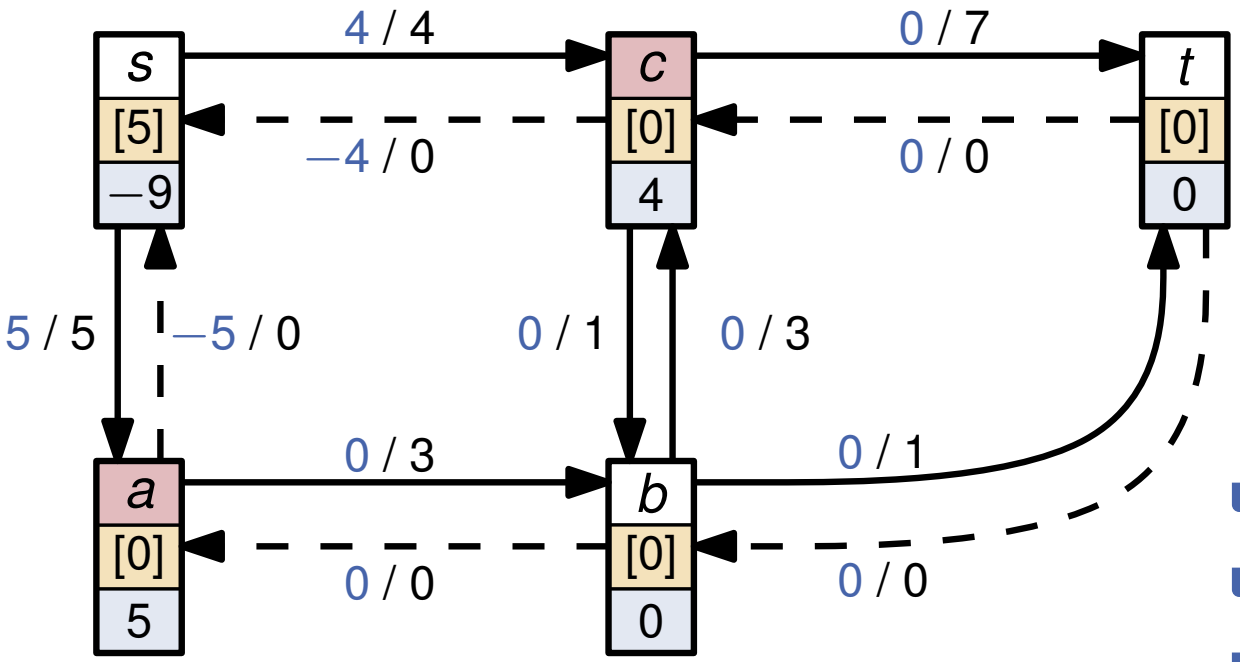
Wähle aktiven Knoten und führe PUSH / RELABEL aus:

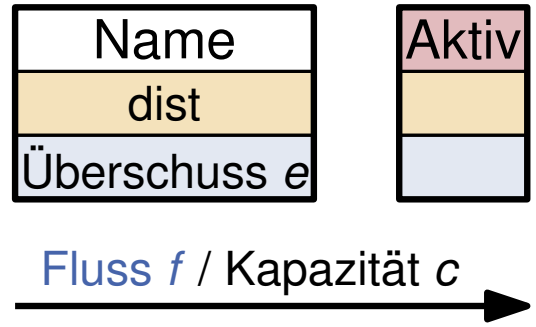
Residualnetzwerk D_f:



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

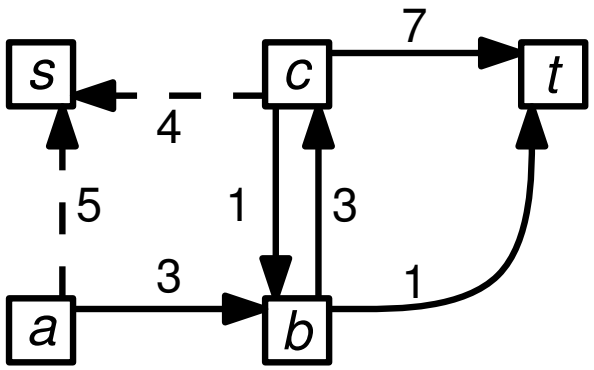
Goldberg & Tarjan





- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

Residualnetzwerk D_f:



Wähle aktiven Knoten und führe PUSH / RELABEL aus:

Bedingungen zum Ausführen von RELABEL für a:

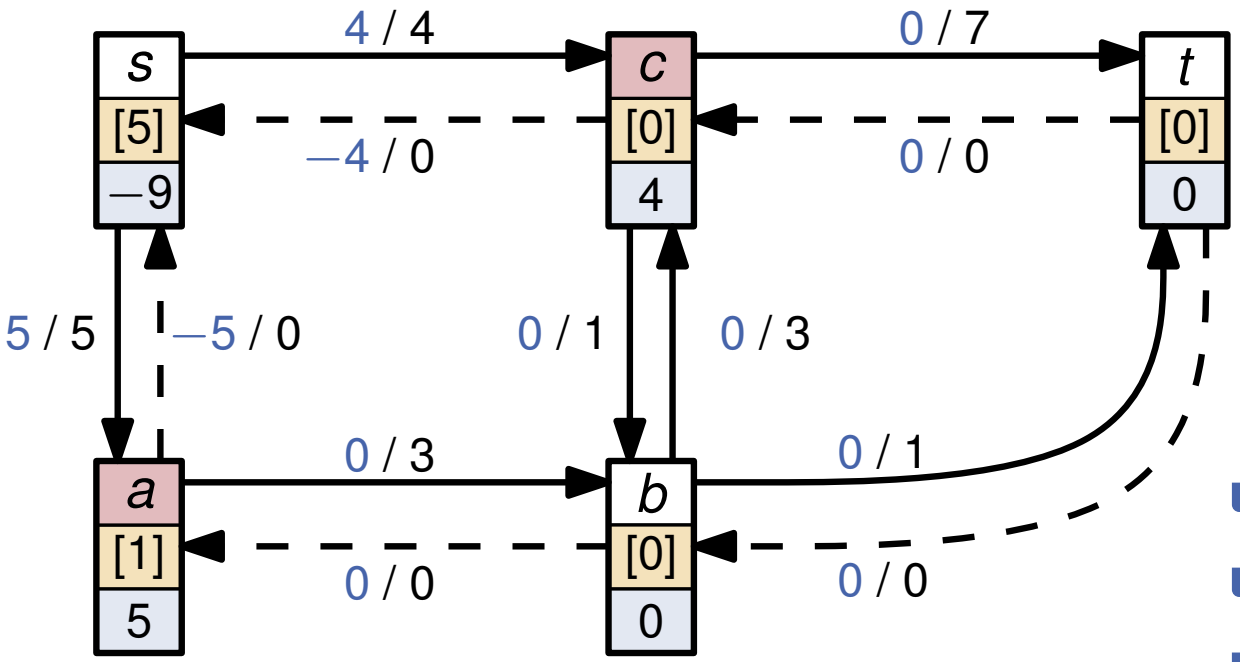
- Für alle Kanten (a, v) in D_f gilt: $\text{dist}(a) \leq \text{dist}(v)$

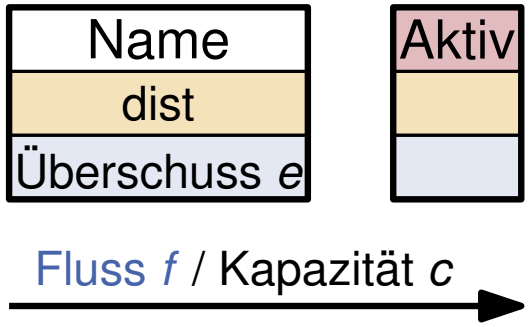
Setze Label $\text{dist}(a)$ auf...

- ∞, falls es in D_f keine Kante (a, v) gibt.
- Minimum von $\text{dist}(v) + 1$ über alle Kanten (a, v) in D_f.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

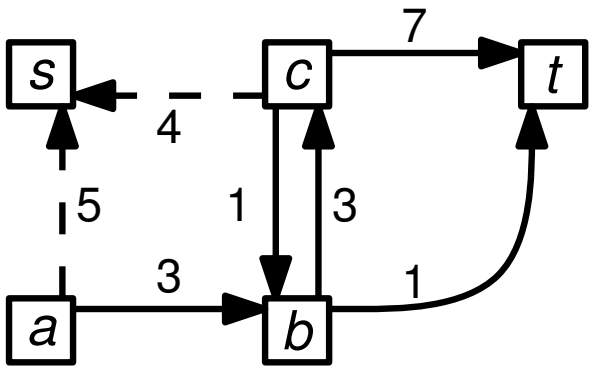
Goldberg & Tarjan





- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

Residualnetzwerk D_f:



Wähle aktiven Knoten und führe PUSH / RELABEL aus:

Bedingungen zum Ausführen von RELABEL für a:

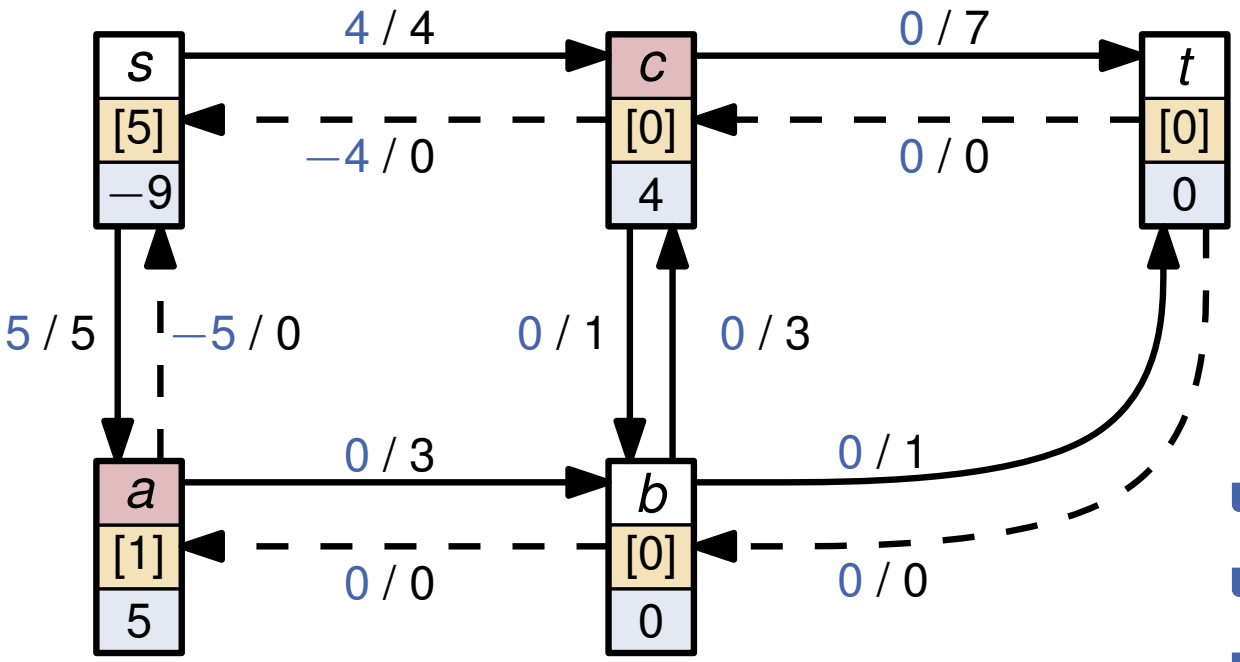
- Für alle Kanten (a, v) in D_f gilt: $\text{dist}(a) \leq \text{dist}(v)$

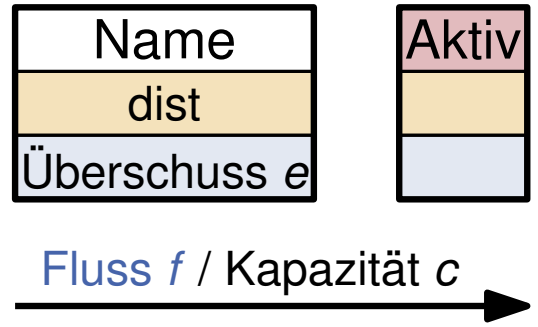
Setze Label $\text{dist}(a)$ auf...

- ∞, falls es in D_f keine Kante (a, v) gibt.
- Minimum von $\text{dist}(v) + 1$ über alle Kanten (a, v) in D_f.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

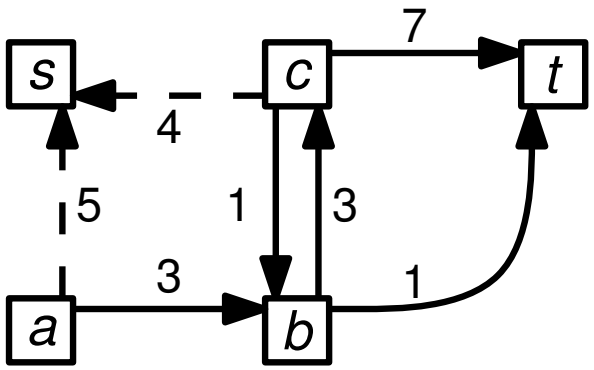
Goldberg & Tarjan





- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

Residualnetzwerk D_f:



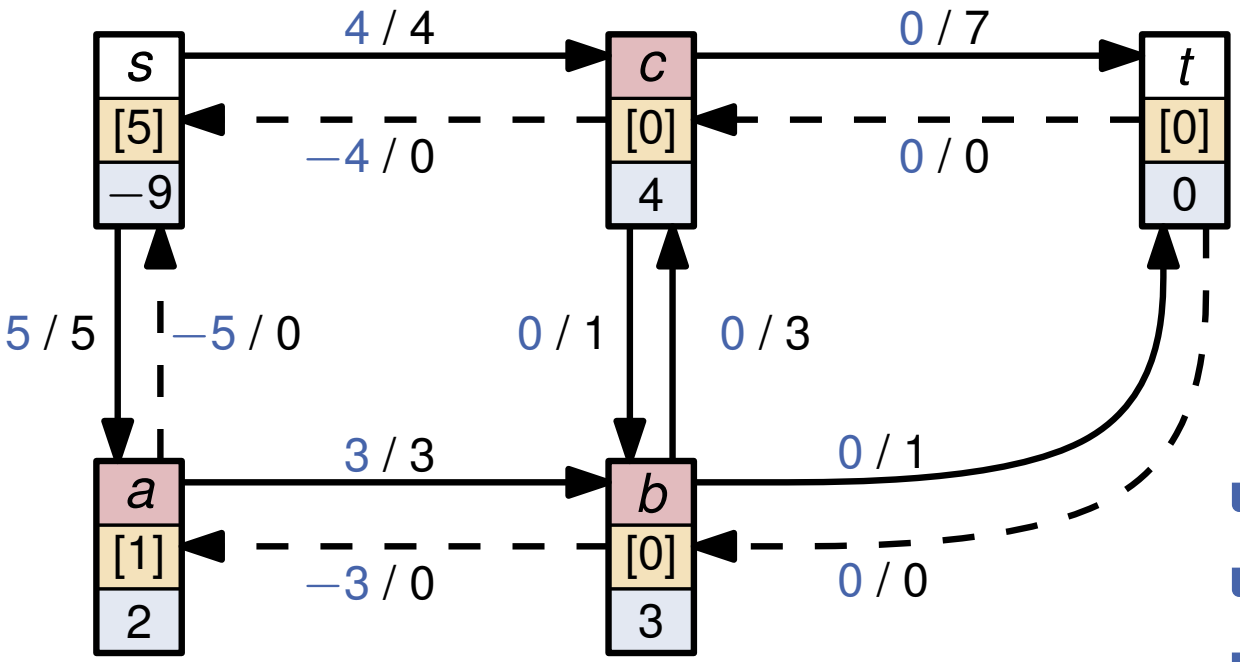
Wähle aktiven Knoten und führe PUSH / RELABEL aus:

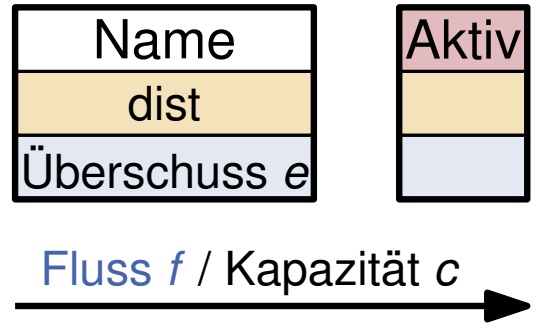
Bedingungen zum Ausführen von PUSH für a:

- D_f enthält Kante (a, v) mit $\text{dist}(a) = \text{dist}(v) + 1$

Erhöhe Fluss auf dieser Kante (a, v) möglichst stark.

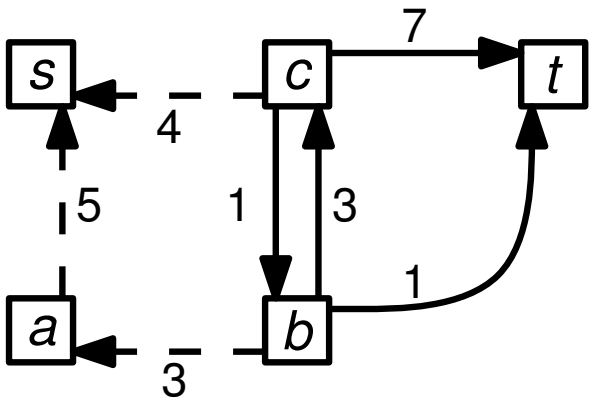
Goldberg & Tarjan





- Führe Gegenkanten ein
- Initialisiere dist und f
- Betrachte Residualnetzwerk

Residualnetzwerk D_f:



Wähle aktiven Knoten und führe PUSH / RELABEL aus:

Bedingungen zum Ausführen von PUSH für a:

- D_f enthält Kante (a, v) mit $\text{dist}(a) = \text{dist}(v) + 1$

Erhöhe Fluss auf dieser Kante (a, v) möglichst stark.

Goldberg & Tarjan

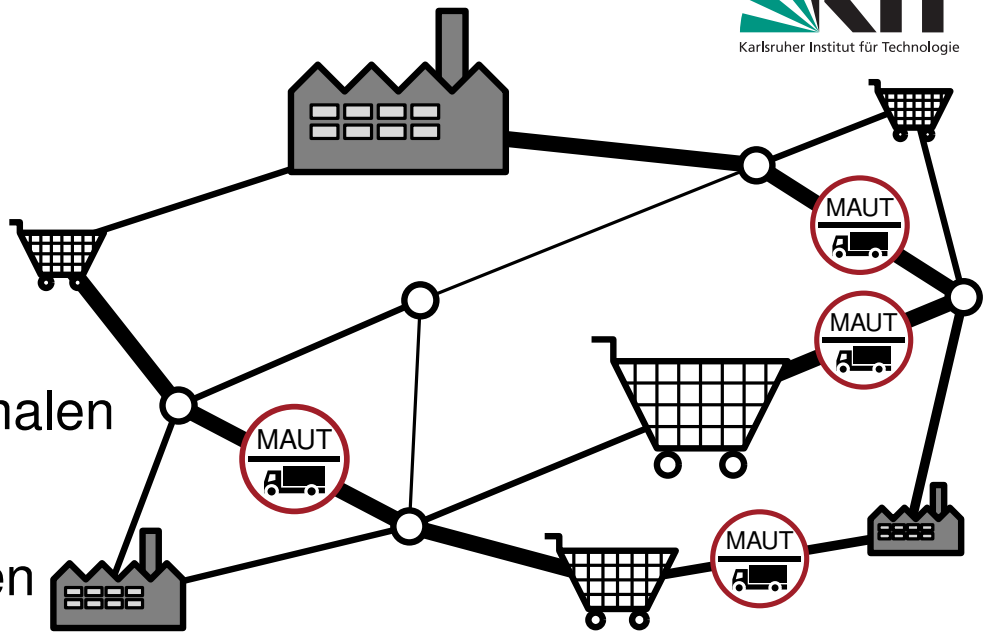
. . . oder auch PUSH – RELABEL

- Was ist die Idee des Goldberg-Tarjan Algos?
- Was versteht man in diesem Kontext unter „Fluss“ und „Residualnetzwerk“?
- Was ist ein Präfluss? Wann ist ein Präfluss ein Fluss?
- Was sind aktive Knoten?
- Was passiert bei den Operationen PUSH bzw. RELABEL?
- Wann sind diese Operationen zulässig?

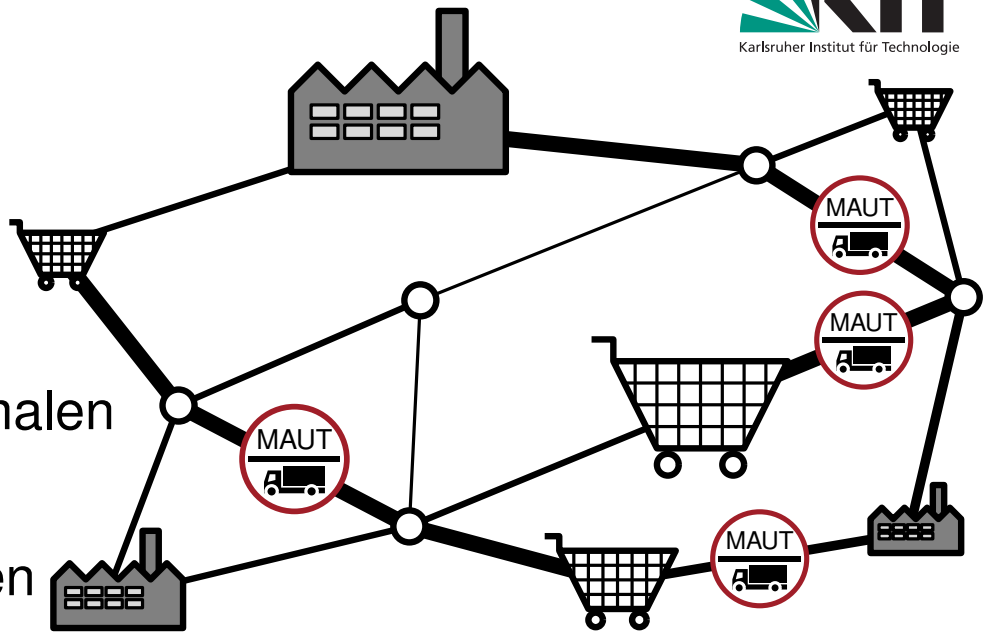
Flüsse mit Kosten

Grundlegendes

- Was ist die Problemstellung? Was sind die Kosten eines Flusses?
- Was ändert sich im Vergleich zu maximalen Flüssen?
- Was hat es mit den Quellen und Senken auf sich? Wie ändert sich die Flusserhaltungsbedingung?



Flüsse mit Kosten



Grundlegendes

- Was ist die Problemstellung? Was sind die Kosten eines Flusses?
- Was ändert sich im Vergleich zu maximalen Flüssen?
- Was hat es mit den Quellen und Senken auf sich? Wie ändert sich die Flusserhaltungsbedingung?

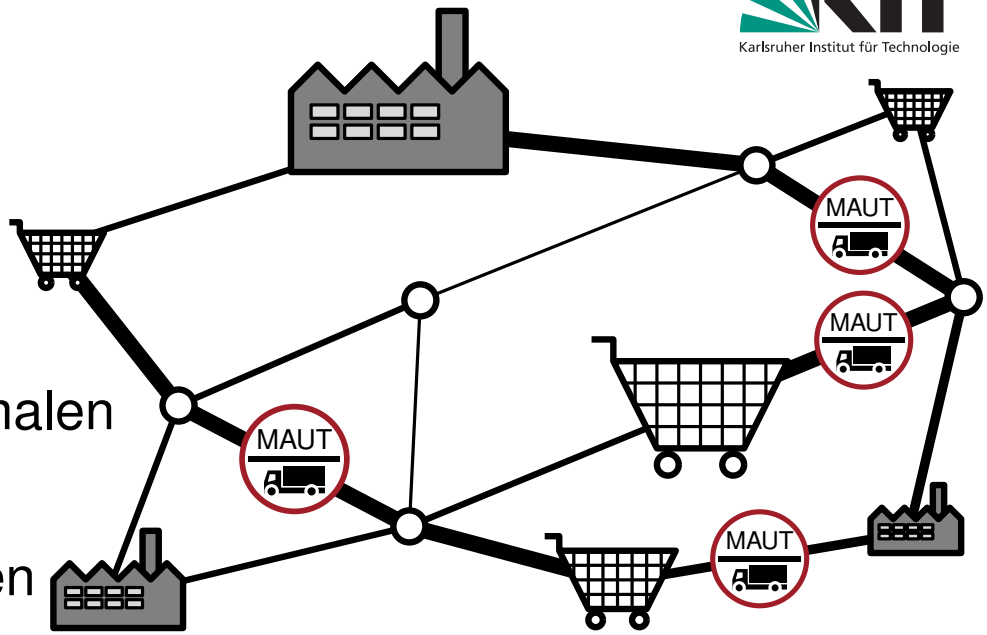
Algorithmisches

- Wie geht man mit mehreren Quellen und Senken um? Wie findet man einen gültigen Fluss?

Flüsse mit Kosten

Grundlegendes

- Was ist die Problemstellung? Was sind die Kosten eines Flusses?
- Was ändert sich im Vergleich zu maximalen Flüssen?
- Was hat es mit den Quellen und Senken auf sich? Wie ändert sich die Flusserhaltungsbedingung?

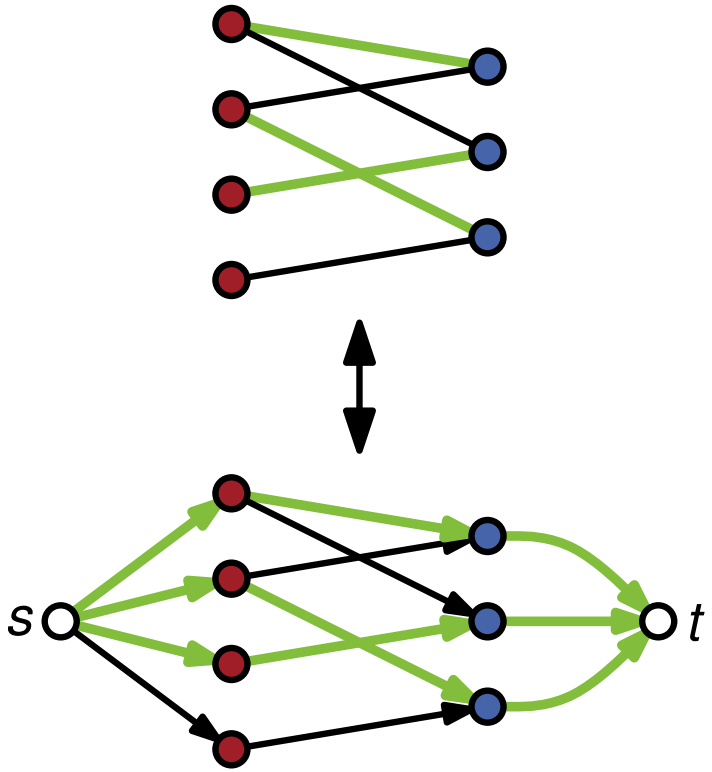


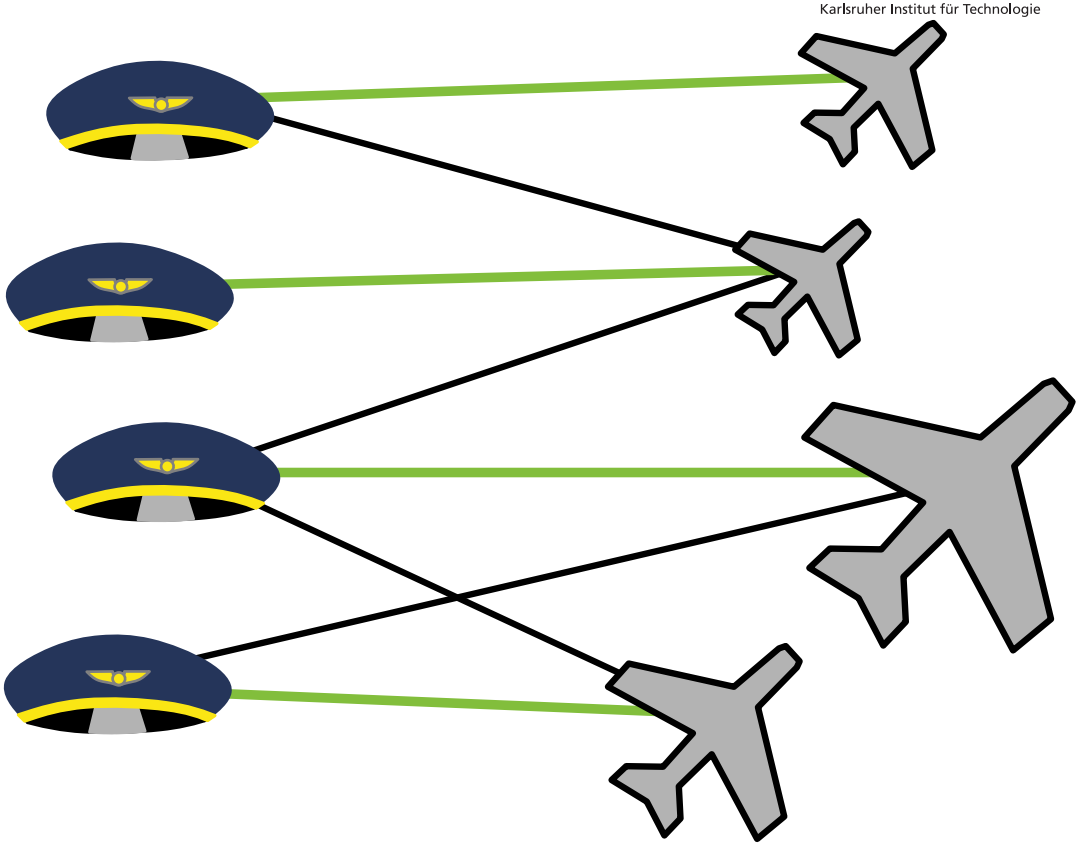
Algorithmisches

- Wie geht man mit mehreren Quellen und Senken um? Wie findet man einen gültigen Fluss?
- Was ist eine erhöhende Zirkulation? Was ist ein erhöhender Kreis?
- Wie können erhöhende Kreise zur Kostenminimierung genutzt werden?

Matchings

- Was ist ein Matching in einem Graphen?





- Was haben maximale Matchings in bipartiten Graphen mit Flüssen zu tun?

(Global) minimale Schnitte

Grundlegendes

- Was ist ein (global) minimaler Schnitt in einem Graphen?
- Wie kann man mit $n - 1$ Flussberechnungen einen minimalen Schnitt finden?
- Warum brauchen wir dann noch einen extra Algorithmus?

(Global) minimale Schnitte

Grundlegendes

- Was ist ein (global) minimaler Schnitt in einem Graphen?
- Wie kann man mit $n - 1$ Flussberechnungen einen minimalen Schnitt finden?
- Warum brauchen wir dann noch einen extra Algorithmus?

Algorithmus von Stoer & Wagner

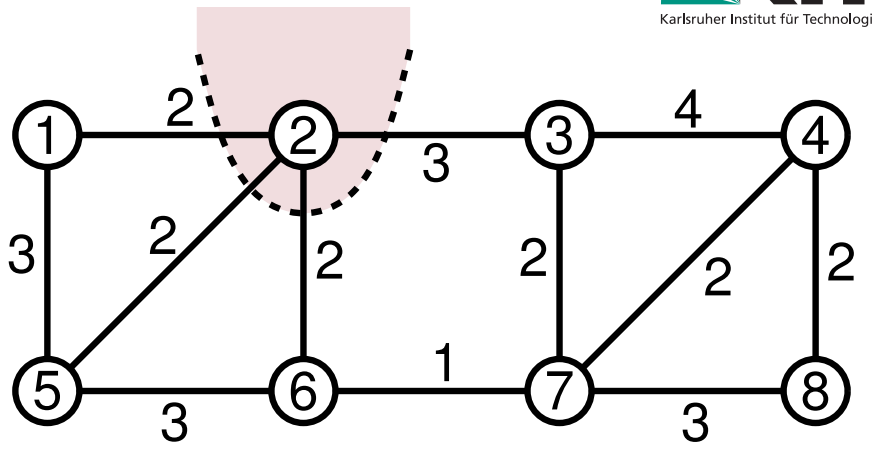
- Wie funktioniert das?

Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)



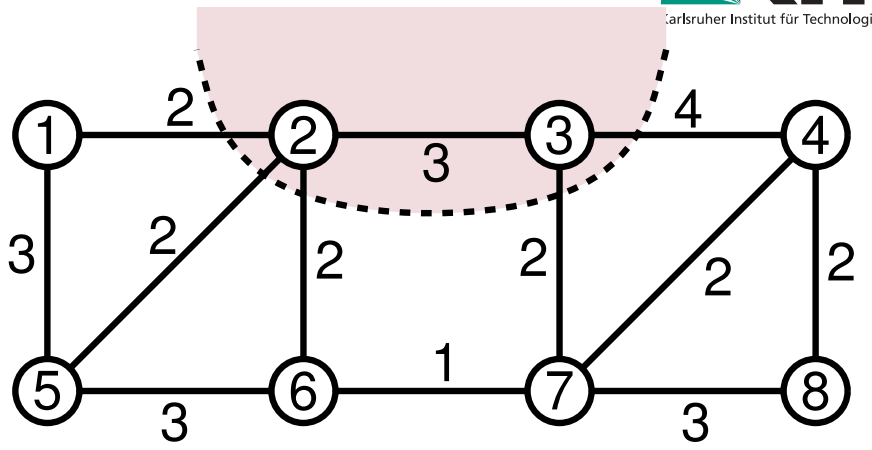
Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

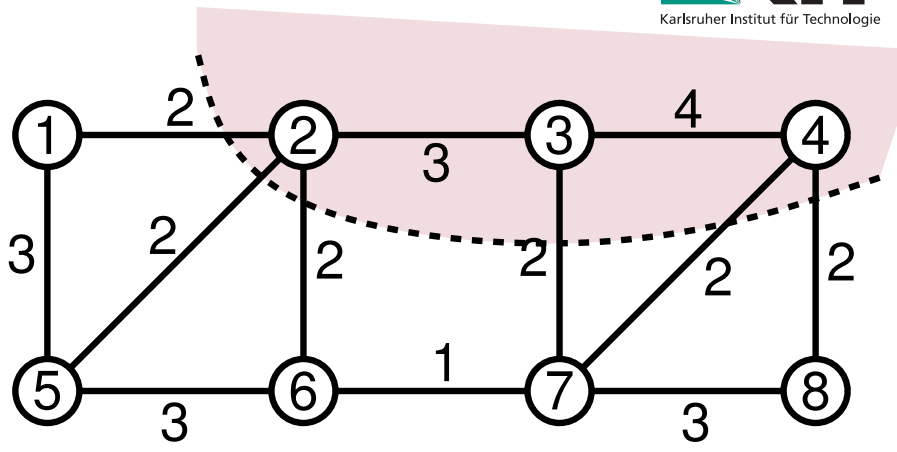
Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)



Algorithmus von Stoer & Wagner – Beispiel

Phase 1

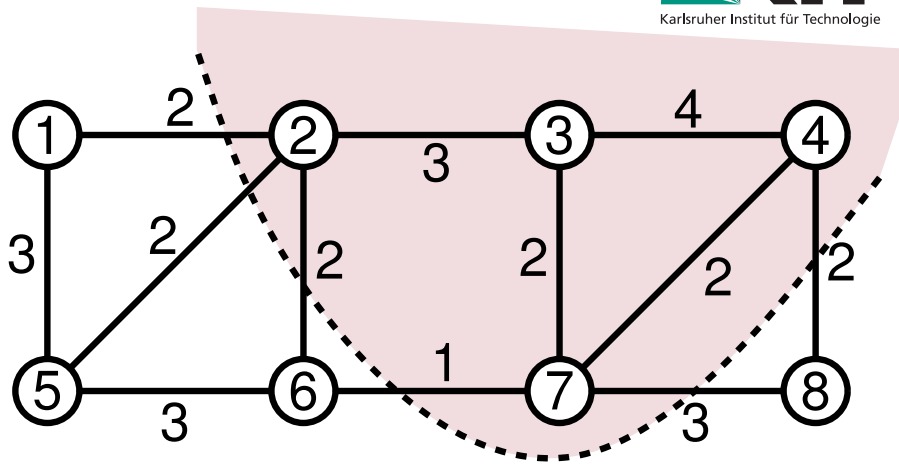
$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

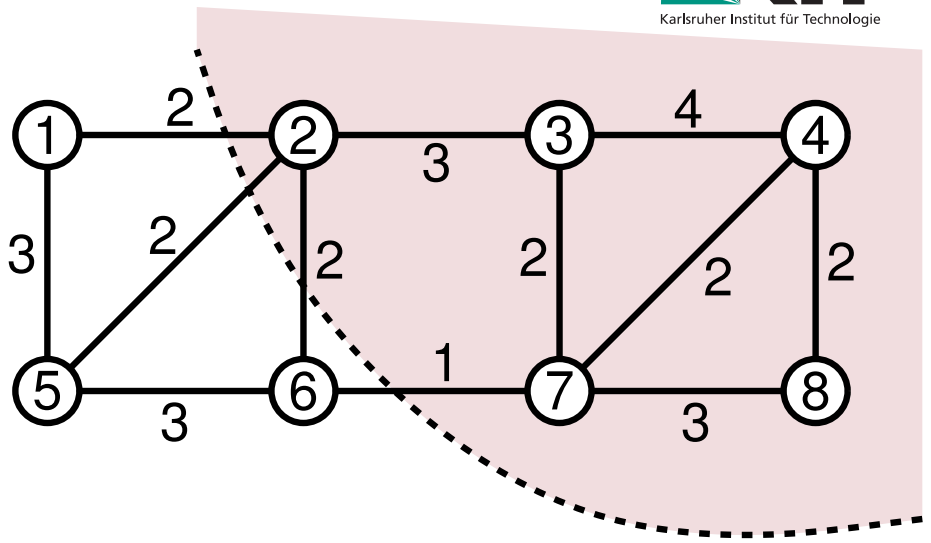
$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$

$S_1 = \{2, 3, 4, 7, 8\}$



Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

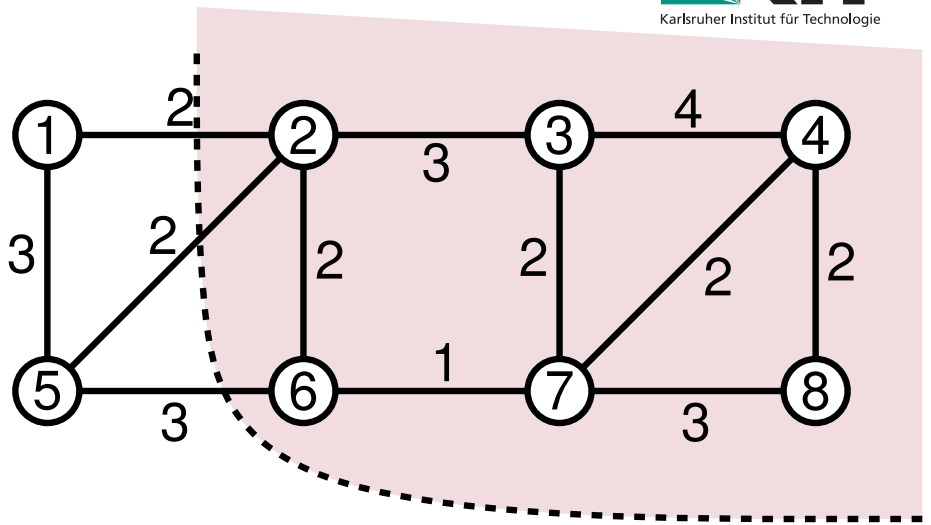
$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$

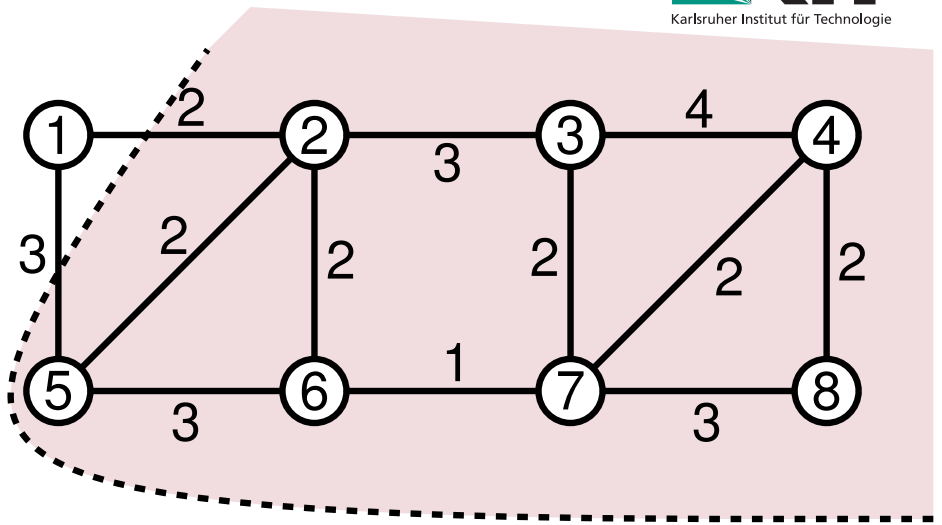
$S_1 = \{2, 3, 4, 7, 8\}$

$S_1 = \{2, 3, 4, 7, 8, 6\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel



Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$

$S_1 = \{2, 3, 4, 7, 8\}$

$S_1 = \{2, 3, 4, 7, 8, 6\}$

$S_1 = \{2, 3, 4, 7, 8, 6, 5\}$

Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$

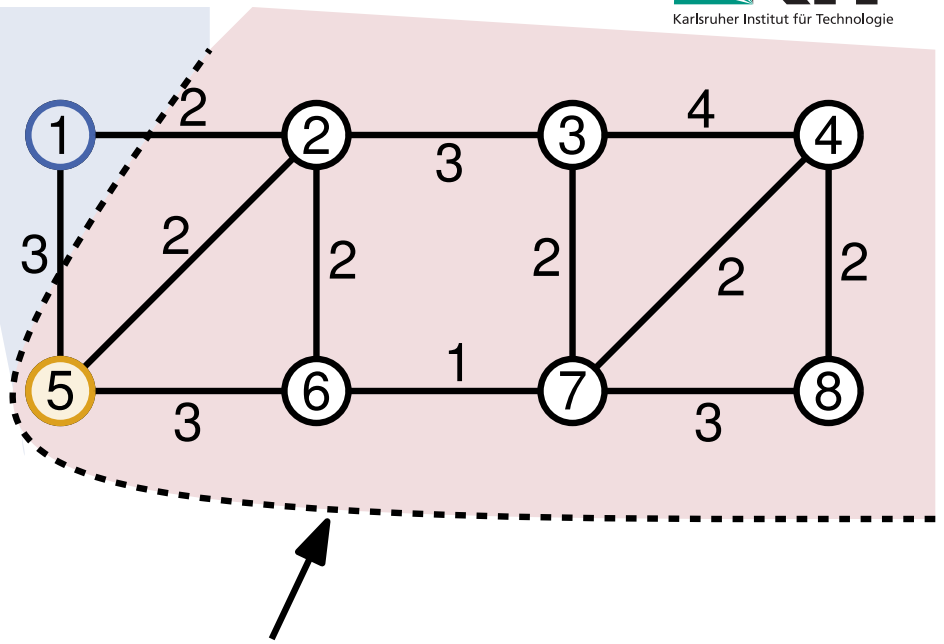
$S_1 = \{2, 3, 4, 7, 8\}$

$S_1 = \{2, 3, 4, 7, 8, 6\}$

$S_1 = \{2, 3, 4, 7, 8, 6, 5\}$

$S_1 = \{2, 3, 4, 7, 8, 6, 5, 1\}$

s t



Schnitt der Phase: $\{V_1 \setminus \{1\}, \{1\}\}$

$\rightarrow$ Gewicht 5

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 1

$G_1 = G$

$S_1 = \{2\}$ (beliebig gewählter Startknoten)

$S_1 = \{2, 3\}$ (3 am stärksten zu $\{2\}$ verbunden)

$S_1 = \{2, 3, 4\}$ (4 am stärksten zu $\{2, 3\}$ verbunden)

$S_1 = \{2, 3, 4, 7\}$

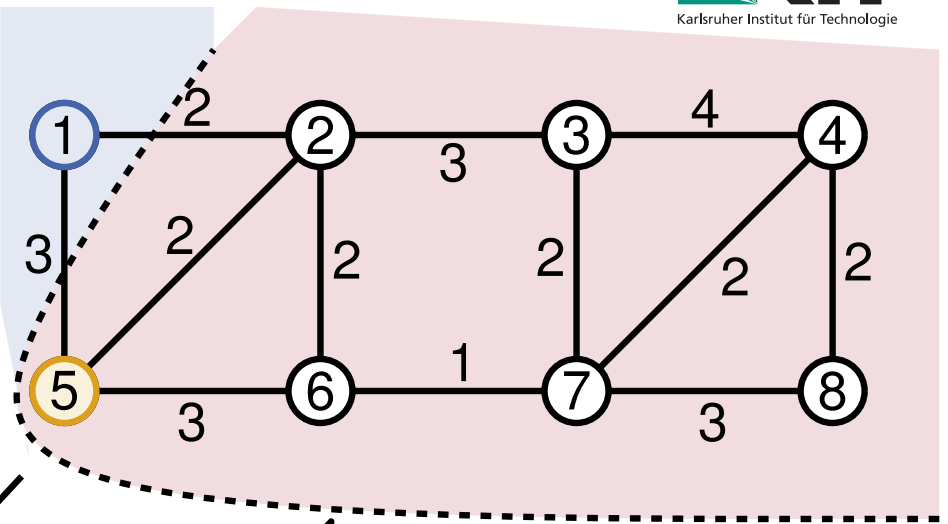
$S_1 = \{2, 3, 4, 7, 8\}$

$S_1 = \{2, 3, 4, 7, 8, 6\}$

$S_1 = \{2, 3, 4, 7, 8, 6, 5\}$

$S_1 = \{2, 3, 4, 7, 8, 6, 5, 1\}$

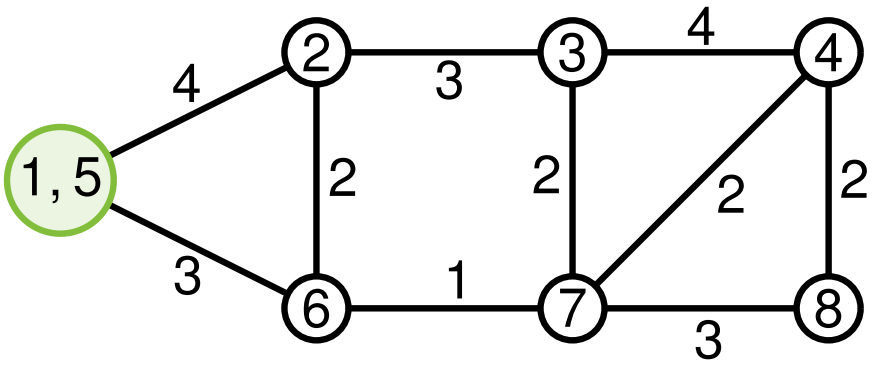
s t



Schnitt der Phase: $\{V_1 \setminus \{1\}, \{1\}\}$

$\rightarrow$ Gewicht 5

Verschmelzen von s und t ergibt G_2



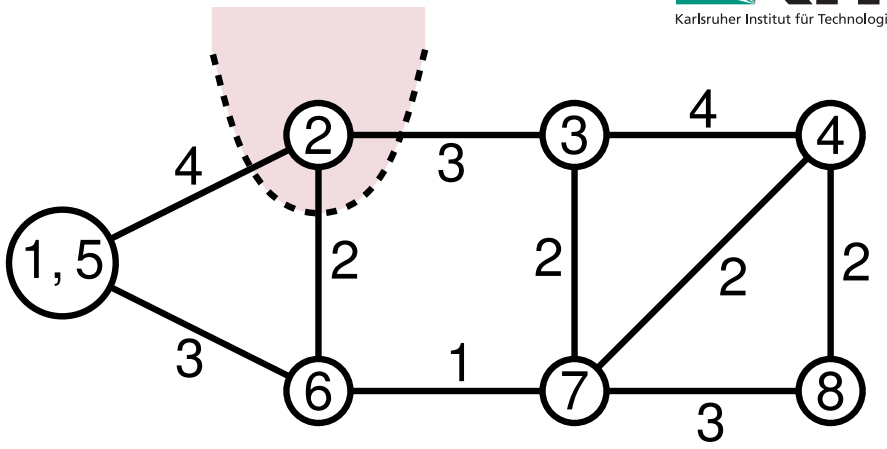
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)



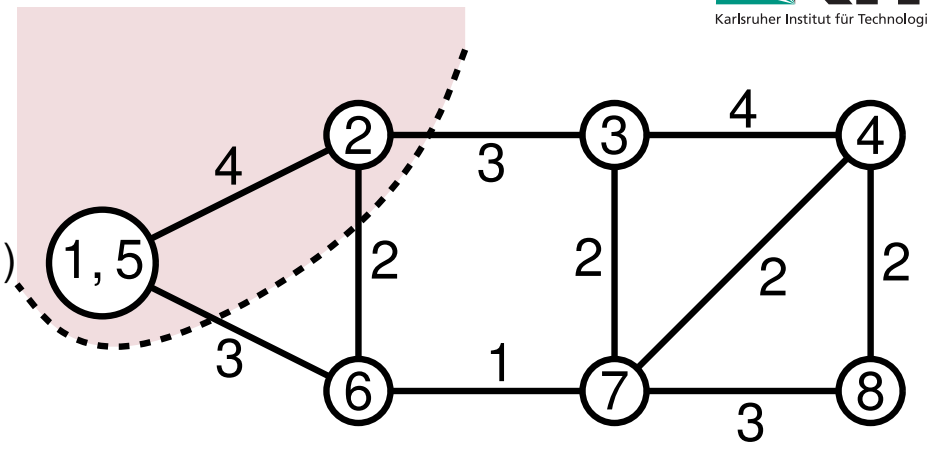
Algorithmus von Stoer & Wagner – Beispiel

Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)

$S_2 = \{2, \{1, 5\}\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

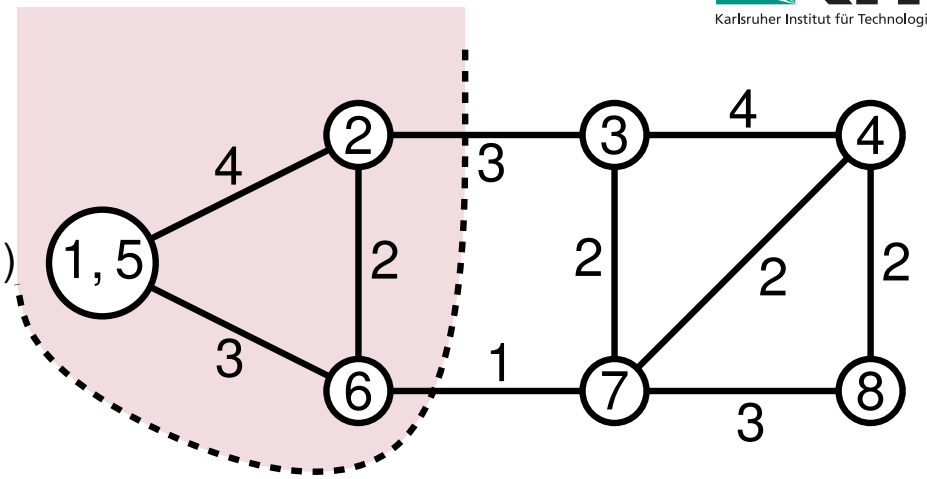
Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)

$S_2 = \{2, \{1, 5\}\}$

$S_2 = \{2, \{1, 5\}, 6\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 2

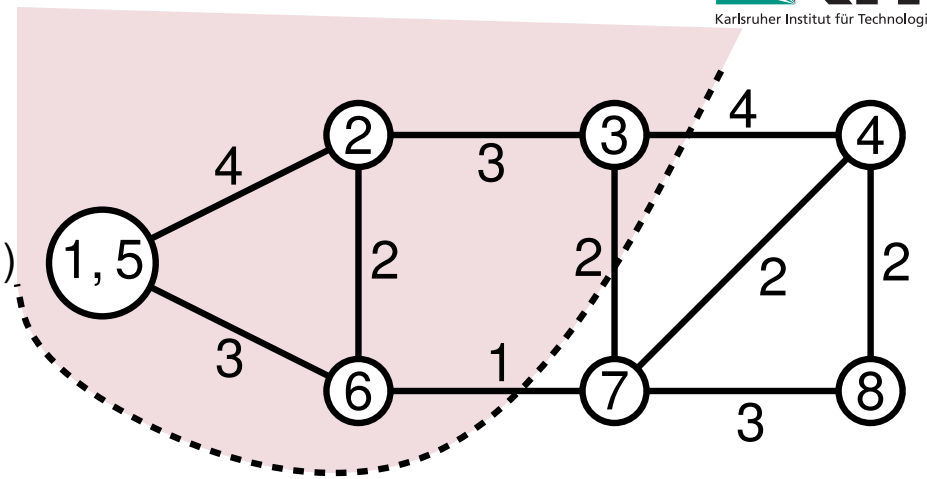
$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)

$S_2 = \{2, \{1, 5\}\}$

$S_2 = \{2, \{1, 5\}, 6\}$

$S_2 = \{2, \{1, 5\}, 6, 3\}$



Algorithmus von Stoer & Wagner – Beispiel

Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

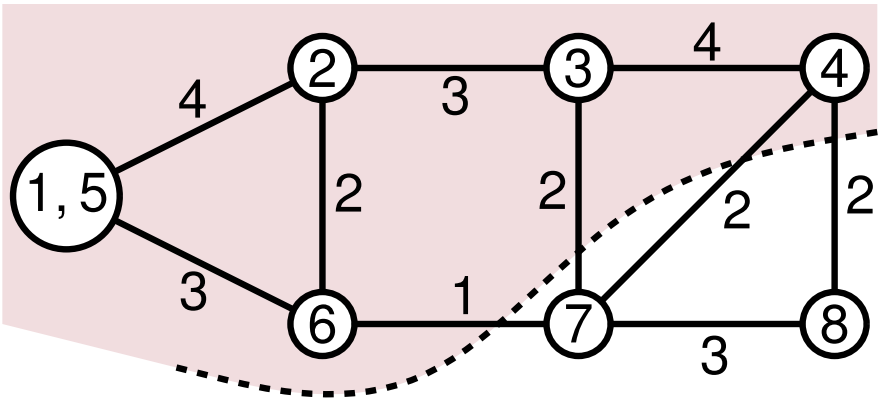
$S_2 = \{2\}$ (beliebig gewählter Startknoten)

$S_2 = \{2, \{1,5\}\}$

$S_2 = \{2, \{1,5\}, 6\}$

$S_2 = \{2, \{1,5\}, 6, 3\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)

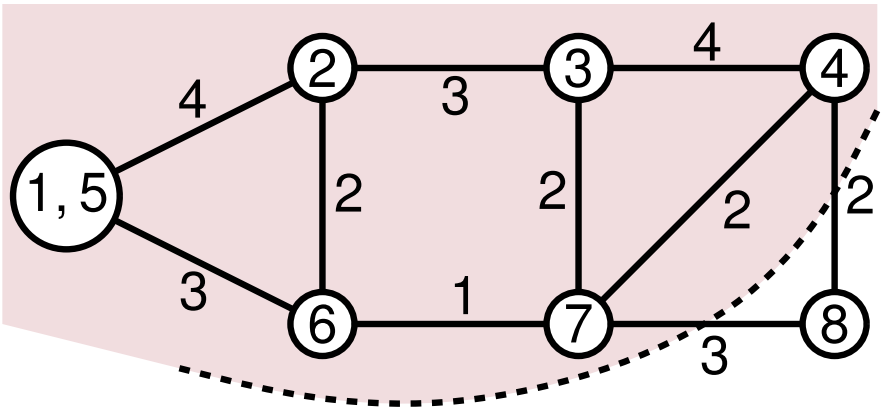
$S_2 = \{2, \{1,5\}\}$

$S_2 = \{2, \{1,5\}, 6\}$

$S_2 = \{2, \{1,5\}, 6, 3\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4, 7\}$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 2

$G_2 = G_1$ mit 1 und 5 verschmolzen

$S_2 = \{2\}$ (beliebig gewählter Startknoten)

$S_2 = \{2, \{1,5\}\}$

$S_2 = \{2, \{1,5\}, 6\}$

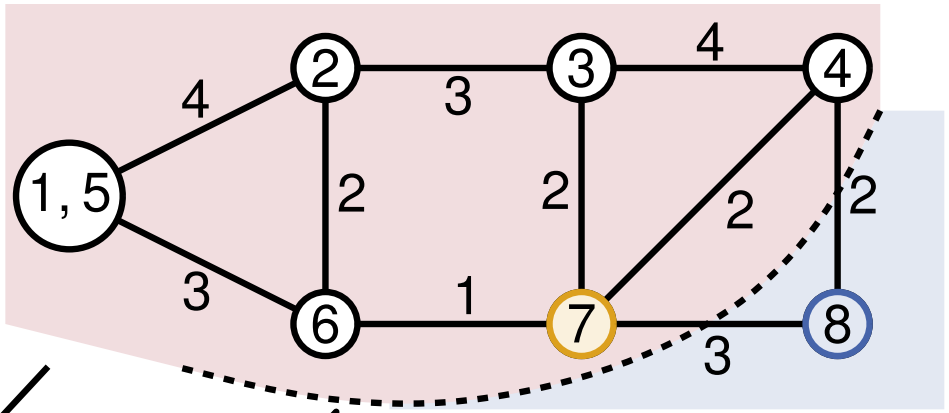
$S_2 = \{2, \{1,5\}, 6, 3\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4, 7\}$

$S_2 = \{2, \{1,5\}, 6, 3, 4, 7, 8\}$

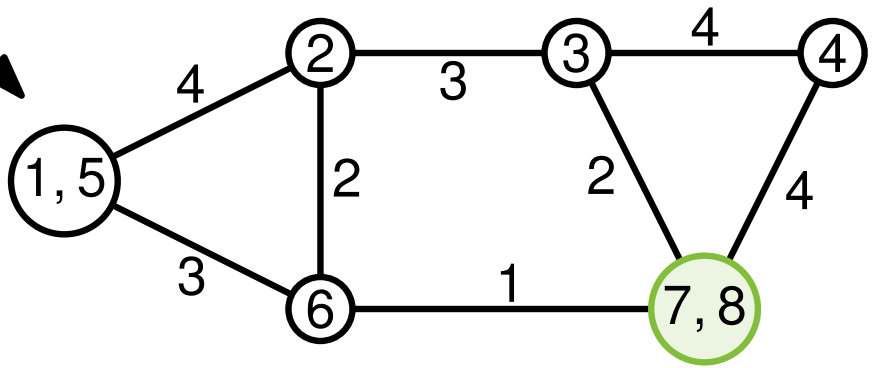
s t



Schnitt der Phase: $\{V_2 \setminus \{8\}, \{8\}\}$

$\rightarrow$ Gewicht 5

Verschmelzen von s und t ergibt G_3



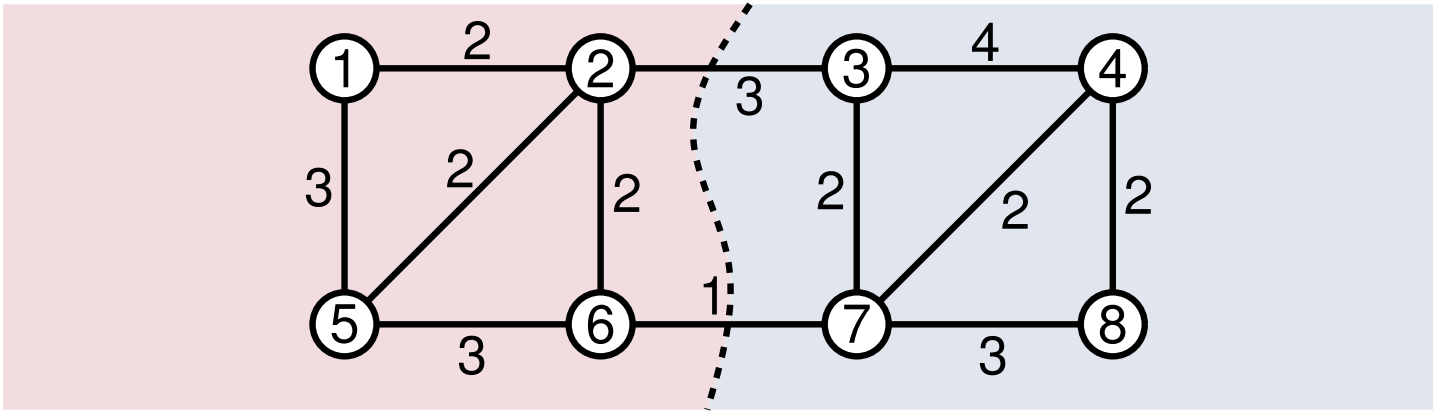
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus von Stoer & Wagner – Beispiel

Phase 1 Schnitt der Phase: $\{V_1 \setminus \{1\}, \{1\}\} \rightarrow$ Gewicht 5
Phase 2 Schnitt der Phase: $\{V_2 \setminus \{8\}, \{8\}\} \rightarrow$ Gewicht 5
Phase 3 Schnitt der Phase: $\{V_3 \setminus \{\{7,8\}\}, \{\{7,8\}\}\} \rightarrow$ Gewicht 7
Phase 4 Schnitt der Phase: $\{V_4 \setminus \{\{4,7,8\}\}, \{\{4,7,8\}\}\} \rightarrow$ Gewicht 7
Phase 5 Schnitt der Phase: $\{V_5 \setminus \{\{3,4,7,8\}\}, \{\{3,4,7,8\}\}\} \rightarrow$ Gewicht 4
Phase 6 Schnitt der Phase: $\{V_6 \setminus \{\{1,5\}\}, \{\{1,5\}\}\} \rightarrow$ Gewicht 7
Phase 7 Schnitt der Phase: $\{V_7 \setminus \{2\}, \{2\}\} \rightarrow$ Gewicht 9

siehe Skript

Der Schnitt aus **Phase 5** ist minimal unter den Schnitten der einzelnen Phasen.
$\Rightarrow$ Der Algorithmus von Stoer & Wagner gibt diesen Schnitt aus.



(Beweis, dass der so bestimmte Schnitt immer ein minimaler Schnitt ist folgt später.)

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

(Global) minimale Schnitte

Grundlegendes

- Was ist ein (global) minimaler Schnitt in einem Graphen?
- Wie kann man mit $n - 1$ Flussberechnungen einen minimalen Schnitt finden?
- Warum brauchen wir dann noch einen extra Algorithmus?

Algorithmus von Stoer & Wagner

- Wie funktioniert das?

(Global) minimale Schnitte

Grundlegendes

- Was ist ein (global) minimaler Schnitt in einem Graphen?
- Wie kann man mit $n - 1$ Flussberechnungen einen minimalen Schnitt finden?
- Warum brauchen wir dann noch einen extra Algorithmus?

Algorithmus von Stoer & Wagner

- Wie funktioniert das?
- Was heißt „am stärksten verbunden“?
- Wie werden Knoten verschmolzen?

(Global) minimale Schnitte

Grundlegendes

- Was ist ein (global) minimaler Schnitt in einem Graphen?
- Wie kann man mit $n - 1$ Flussberechnungen einen minimalen Schnitt finden?
- Warum brauchen wir dann noch einen extra Algorithmus?

Algorithmus von Stoer & Wagner

- Wie funktioniert das?
- Was heißt „am stärksten verbunden“?
- Wie werden Knoten verschmolzen?
- Was hat es mit s und t und dem Schnitt jeder Phase auf sich?
- Warum funktioniert der Algorithmus?

Matroide

- Was ist ein Unabhängigkeitssystem?
- Wann ist ein Unabhängigkeitssystem ein Matroid?

Matroide

- Was ist ein Unabhängigkeitssystem?
- Wann ist ein Unabhängigkeitssystem ein Matroid?
- Was ist eine Basis eines Unabhängigkeitssystems bzw. Matroids?
- Wie kann man in Matroiden maximale unabhängige Mengen, bzw. minimale Basen finden?
- Wie funktioniert die Greedy-Methode?

Kreisbasen

Grundlegendes

- Was ist ein Kreis im Kontext von Kreisräumen?
- Wie ist die Addition auf Kreisen definiert und warum liefert die Menge aller Kreise einen Vektorraum?

Kreisbasen

Grundlegendes

- Was ist ein Kreis im Kontext von Kreisräumen?
- Wie ist die Addition auf Kreisen definiert und warum liefert die Menge aller Kreise einen Vektorraum?
- Was ist ein Fundamentalkreis? Was ist eine Fundamentalbasis des Kreisraums? Ist jede Kreisbasis eine Fundamentalbasis?
- Was ist die Dimension eines Kreisraums?

Kreisbasen

Grundlegendes

- Was ist ein Kreis im Kontext von Kreisräumen?
- Wie ist die Addition auf Kreisen definiert und warum liefert die Menge aller Kreise einen Vektorraum?
- Was ist ein Fundamentalkreis? Was ist eine Fundamentalbasis des Kreisraums? Ist jede Kreisbasis eine Fundamentalbasis?
- Was ist die Dimension eines Kreisraums?

Algorithmen zur Berechnung einer Kreisbasis minimalen Gewichts

- Was haben Kreisräume mit Matroiden zu tun?
- Warum benutzt man nicht einfach die Greedy-Methode zur Berechnung einer minimalen Kreisbasis?
- Was machen die Algorithmen von Horton bzw. de Pina? Wie schnell sind sie?

Randomisierte Algorithmen

Grundlegendes

- Wann ist ein Algorithmus randomisiert?
- Was unterscheidet einen Las Vegas von einem Monte Carlo Algorithmus?
- Was ist ein beidseitiger bzw. einseitiger Fehler?
- Wie sind die Probleme MAX SAT und MAX CUT definiert?

Randomisierte Algorithmen

Grundlegendes

- Wann ist ein Algorithmus randomisiert?
- Was unterscheidet einen Las Vegas von einem Monte Carlo Algorithmus?
- Was ist ein beidseitiger bzw. einseitiger Fehler?
- Wie sind die Probleme MAX SAT und MAX CUT definiert?

Algorithmisches

- Wie kann man die Fehlerwahrscheinlichkeit bei einem Monte Carlo Algorithmus analysieren (Beispiel MINCUT) und wie kann man das ggf. verbessern?

Randomisierte Algorithmen

Grundlegendes

- Wann ist ein Algorithmus randomisiert?
- Was unterscheidet einen Las Vegas von einem Monte Carlo Algorithmus?
- Was ist ein beidseitiger bzw. einseitiger Fehler?
- Wie sind die Probleme MAX SAT und MAX CUT definiert?

Algorithmisches

- Wie kann man die Fehlerwahrscheinlichkeit bei einem Monte Carlo Algorithmus analysieren (Beispiel MINCUT) und wie kann man das ggf. verbessern?
- Warum liefert RANDOM SAT erwartet eine 2-Approximation?

Randomisierte Algorithmen

Grundlegendes

- Wann ist ein Algorithmus randomisiert?
- Was unterscheidet einen Las Vegas von einem Monte Carlo Algorithmus?
- Was ist ein beidseitiger bzw. einseitiger Fehler?
- Wie sind die Probleme MAX SAT und MAX CUT definiert?

Algorithmisches

- Wie kann man die Fehlerwahrscheinlichkeit bei einem Monte Carlo Algorithmus analysieren (Beispiel MINCUT) und wie kann man das ggf. verbessern?
- Warum liefert RANDOM SAT erwartet eine 2-Approximation?
- Wie kann MAX CUT als quadratisches Programm formuliert werden? Und was hilft das?

Algorithmische Geometrie

Grundlegendes

- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmische Geometrie

Grundlegendes

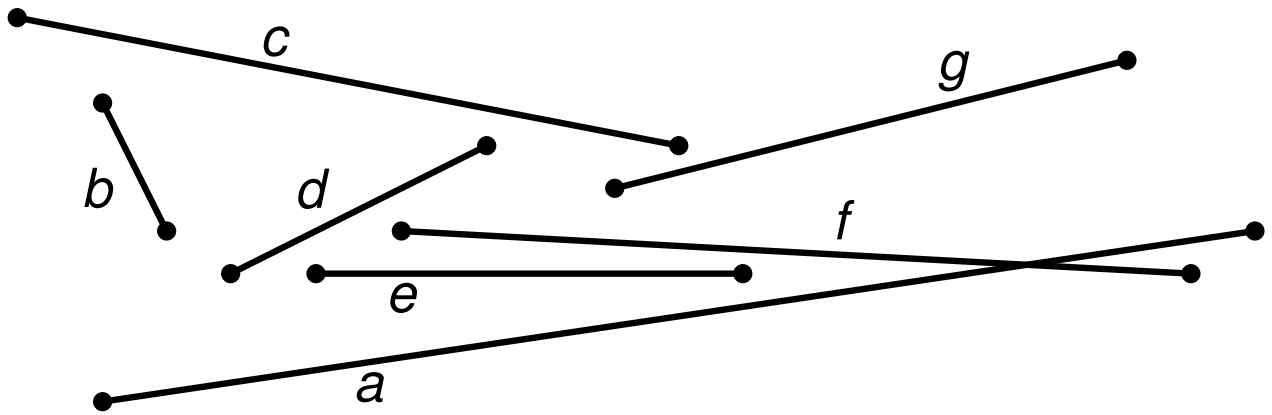
- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmisches

- Was ist die Grundidee eines Sweep-Line Algorithmus? Was speichert man im Laufe des Algorithmus?
- Wie und mit welcher Laufzeit kann man Testen, ob es unter einer Menge von Strecken ein Paar gibt, das sich schneidet?

Sweep-Line Algorithmus – Beispiel

Sweep-Line Zustand

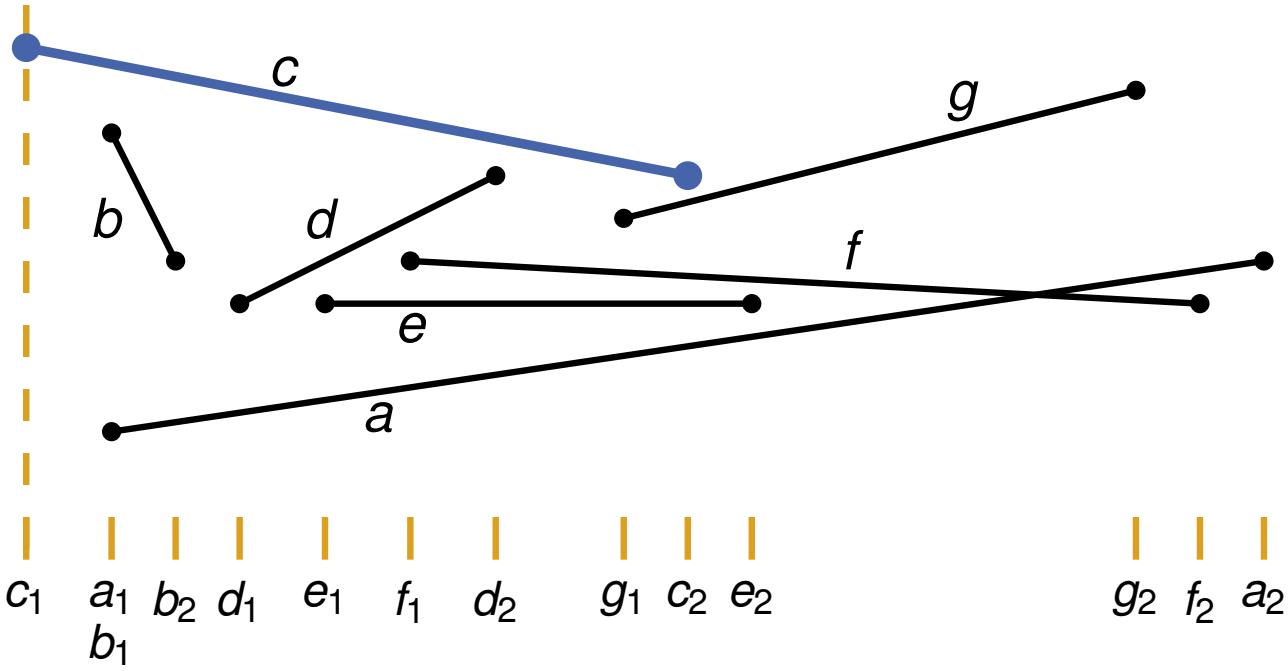


Event-Point Schedule

Sweep-Line Algorithmus – Beispiel

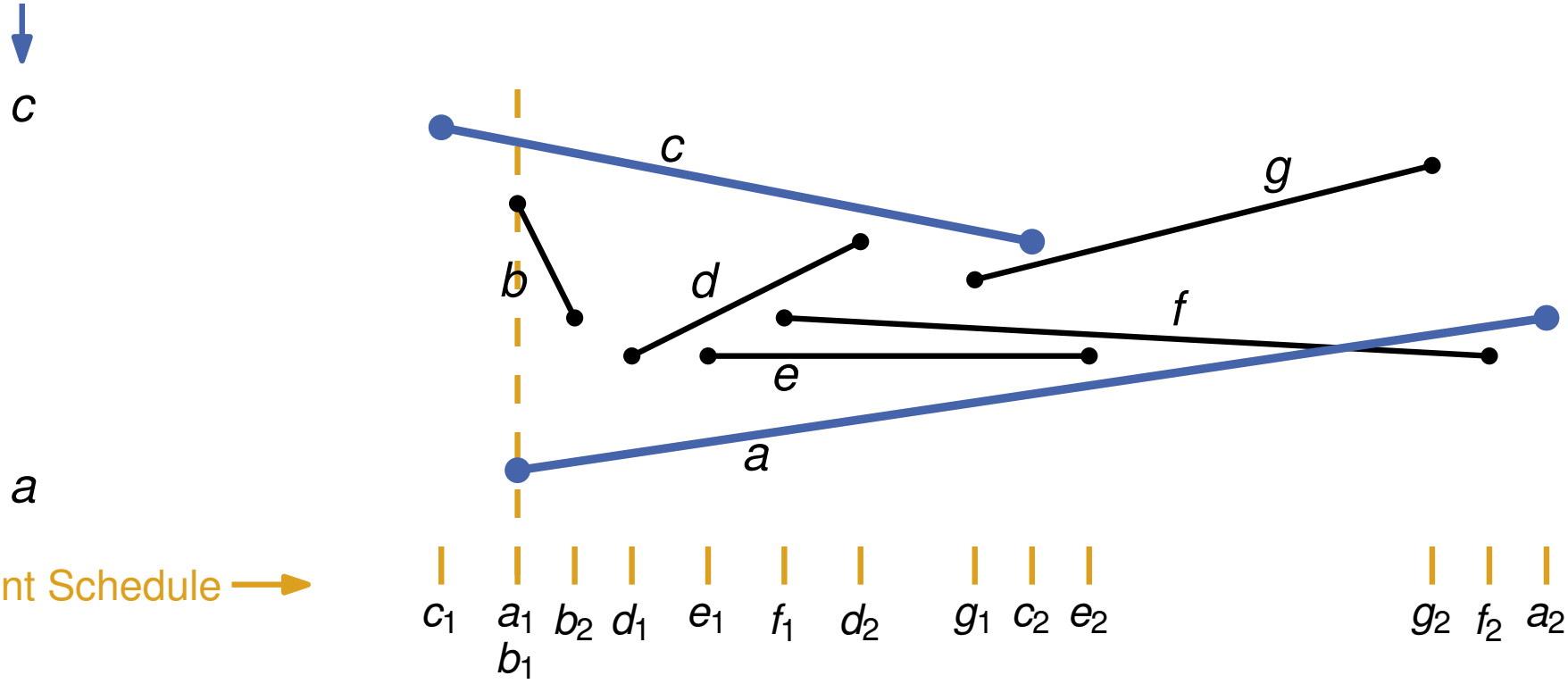
Sweep-Line Zustand

+ c



Sweep-Line Algorithmus – Beispiel

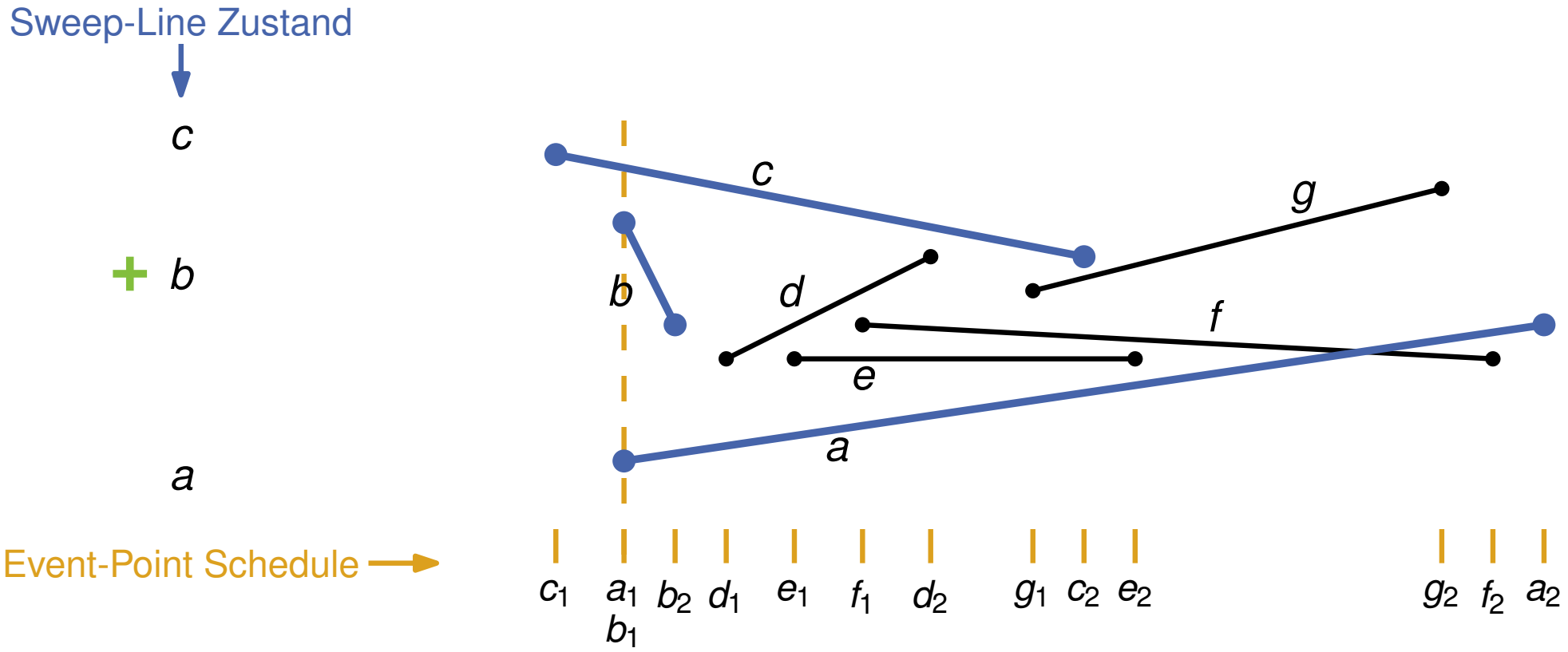
Sweep-Line Zustand



a schneidet *c* nicht

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

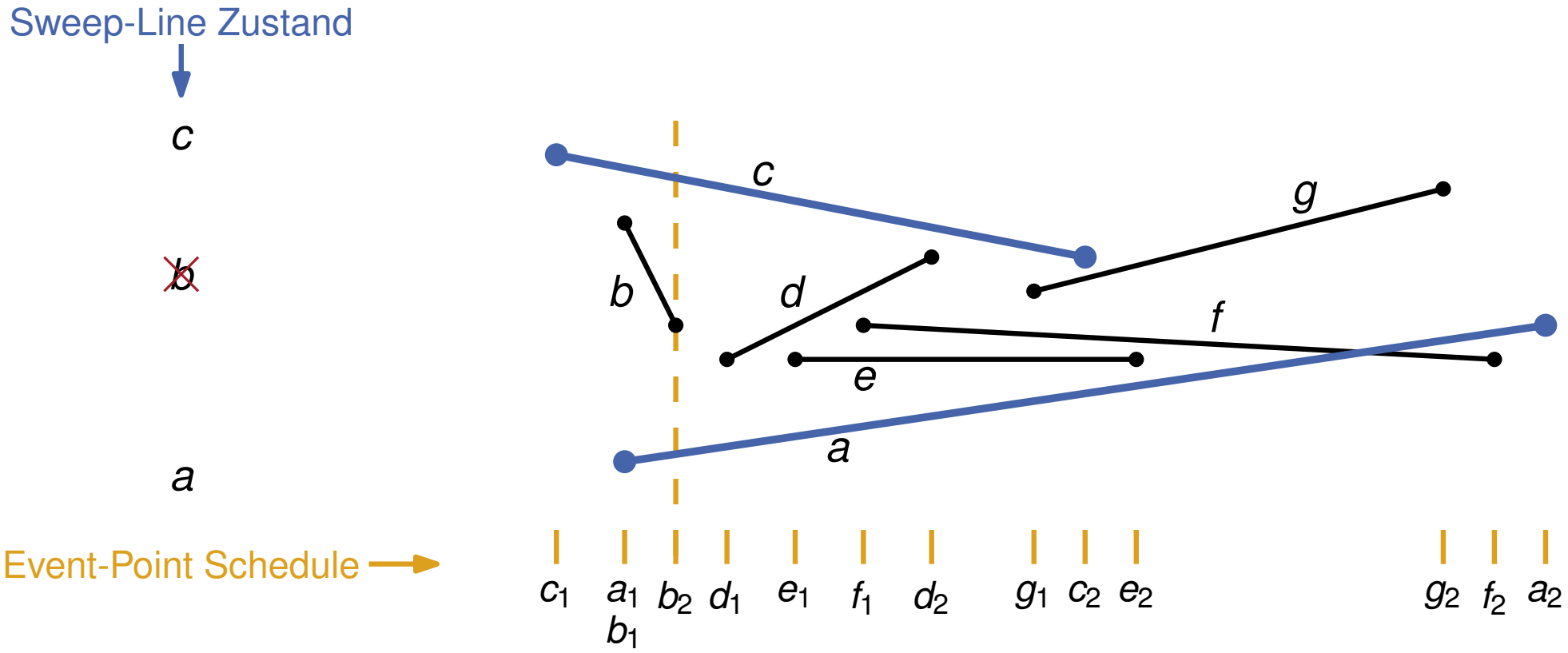
Sweep-Line Algorithmus – Beispiel



b schneidet weder *a* noch *c*

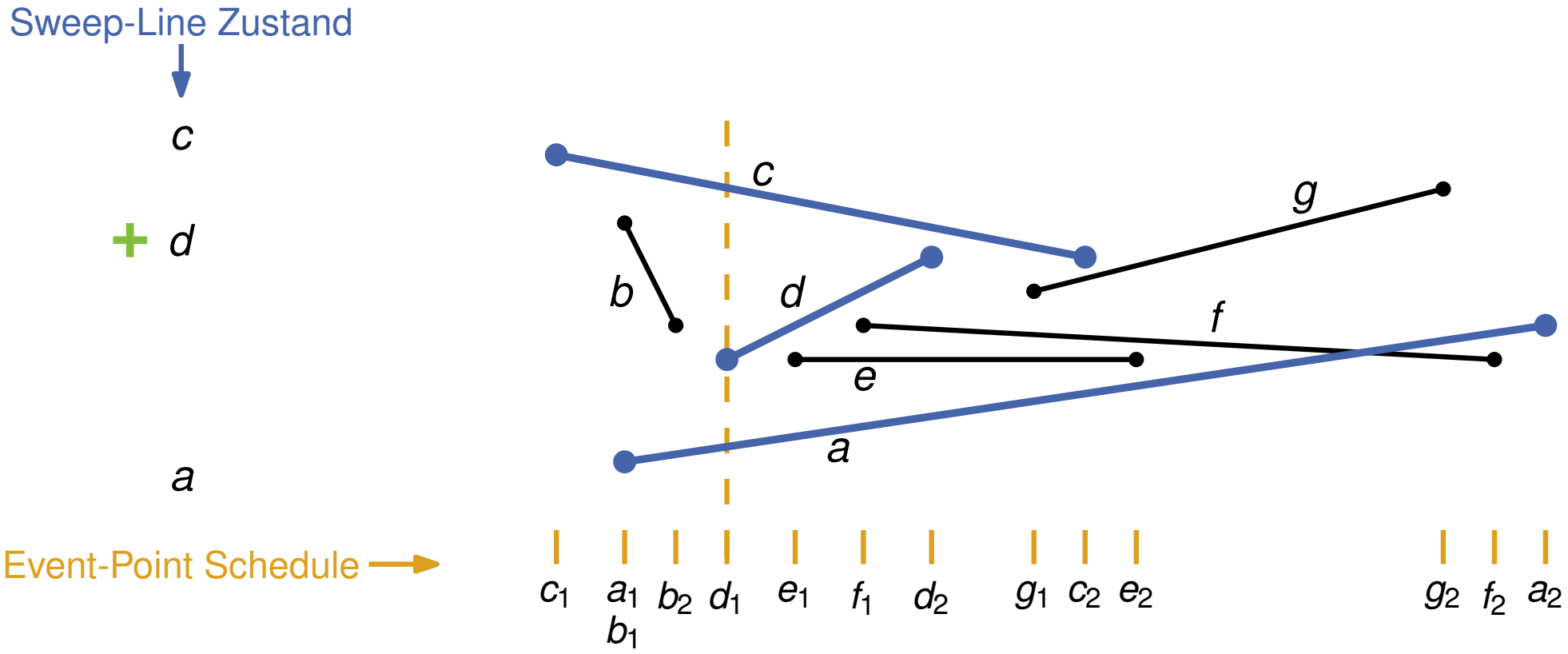
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Sweep-Line Algorithmus – Beispiel



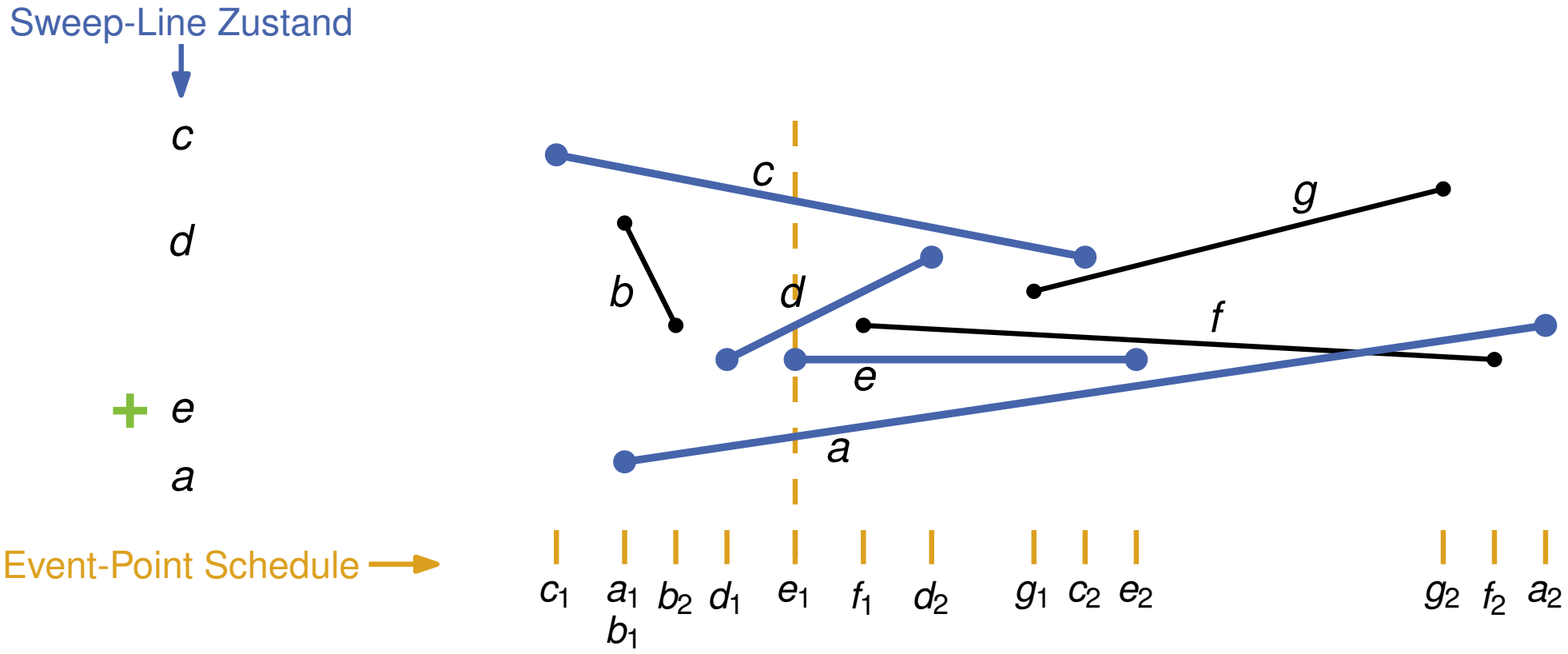
a schneidet *c* nicht

Sweep-Line Algorithmus – Beispiel



d schneidet weder *a* noch *c*

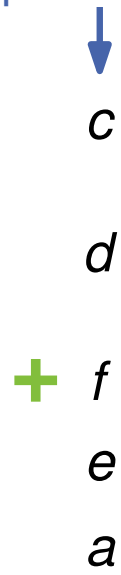
Sweep-Line Algorithmus – Beispiel

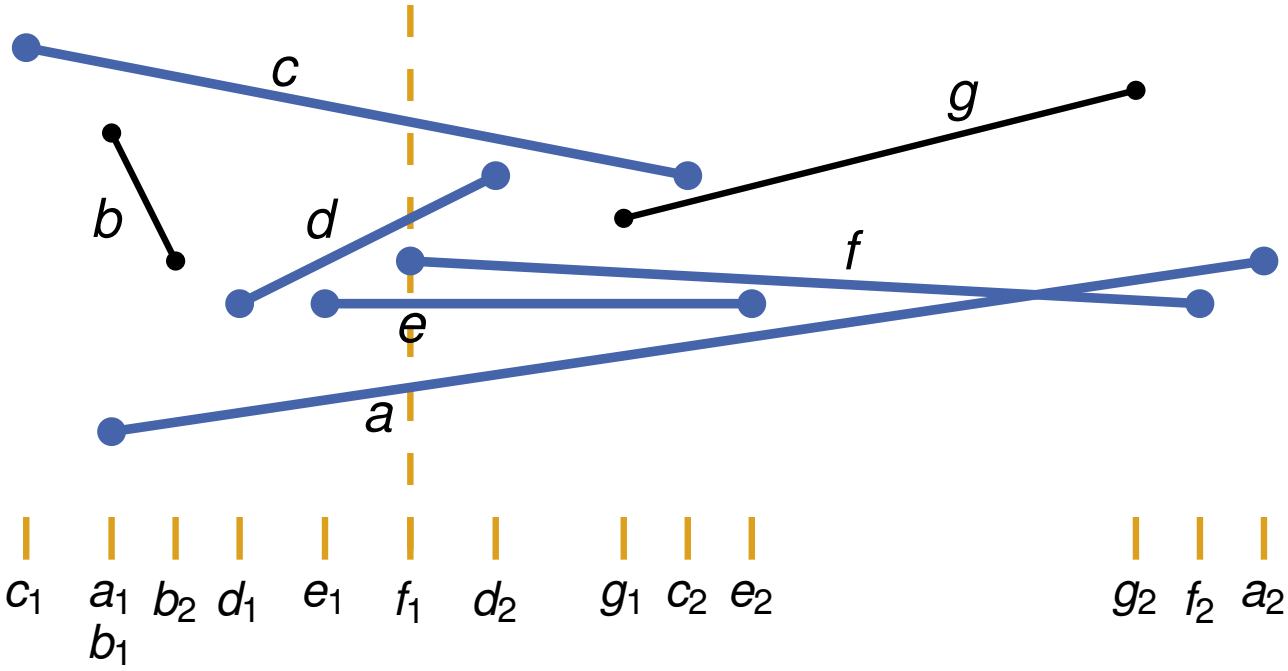


e schneidet weder *d* noch *a*

Sweep-Line Algorithmus – Beispiel

Sweep-Line Zustand





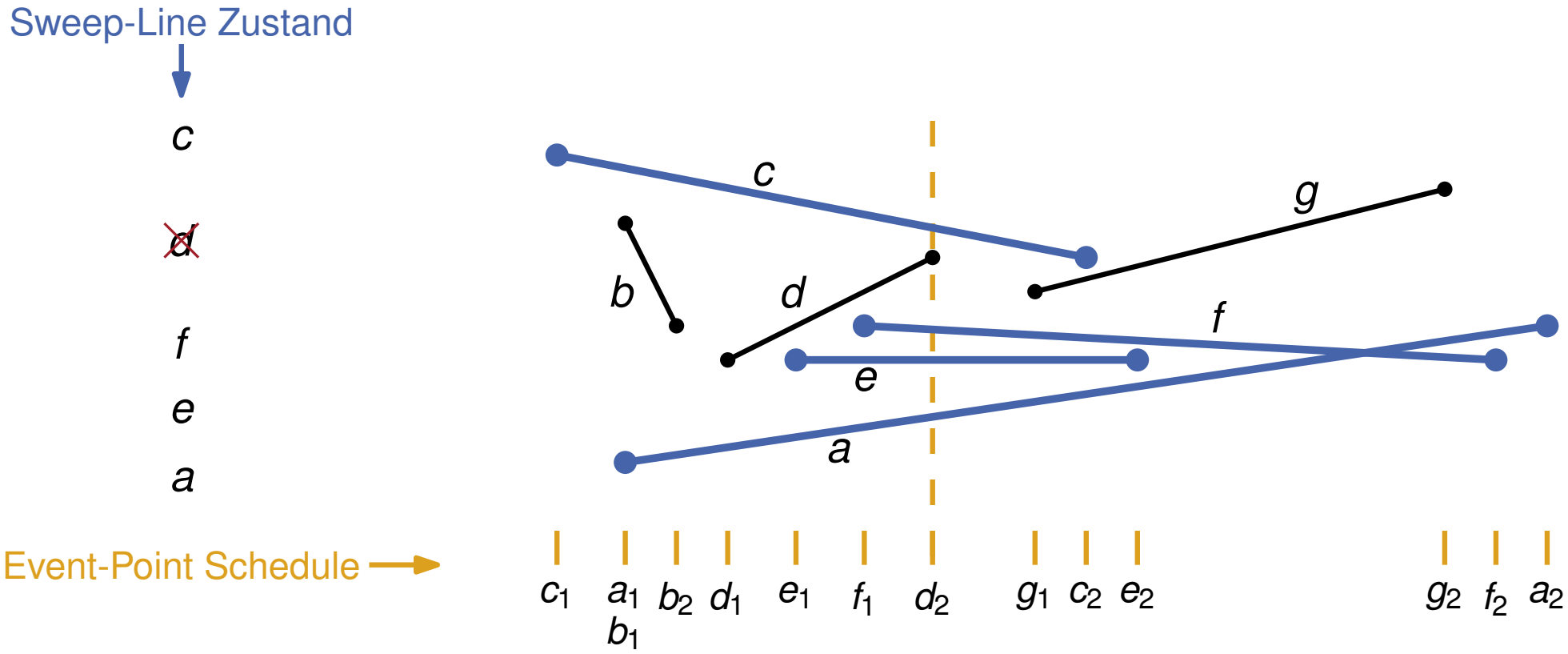
Event-Point Schedule →

c_1 a_1 b_2 d_1 e_1 f_1 d_2 g_1 c_2 e_2 g_2 f_2 a_2
b_1

f schneidet weder *d* noch *e*

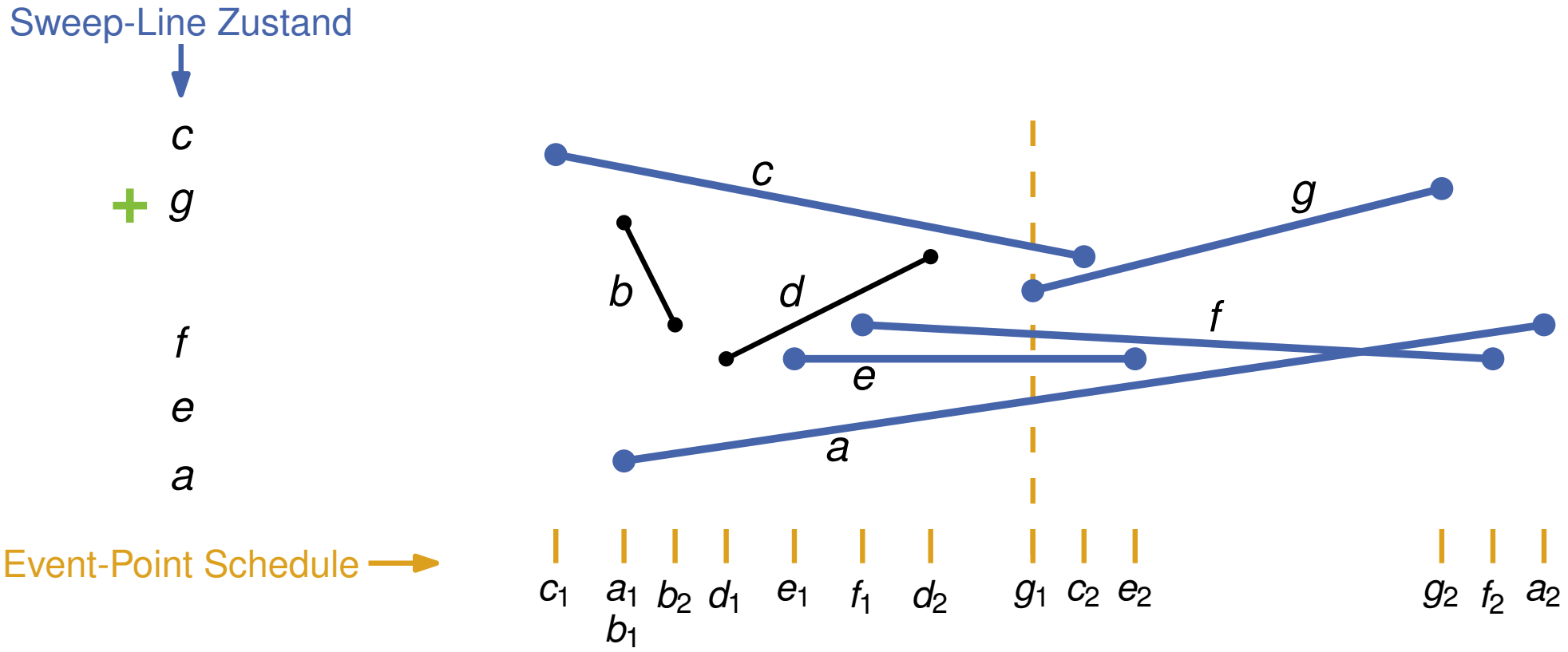
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Sweep-Line Algorithmus – Beispiel



f schneidet *c* nicht

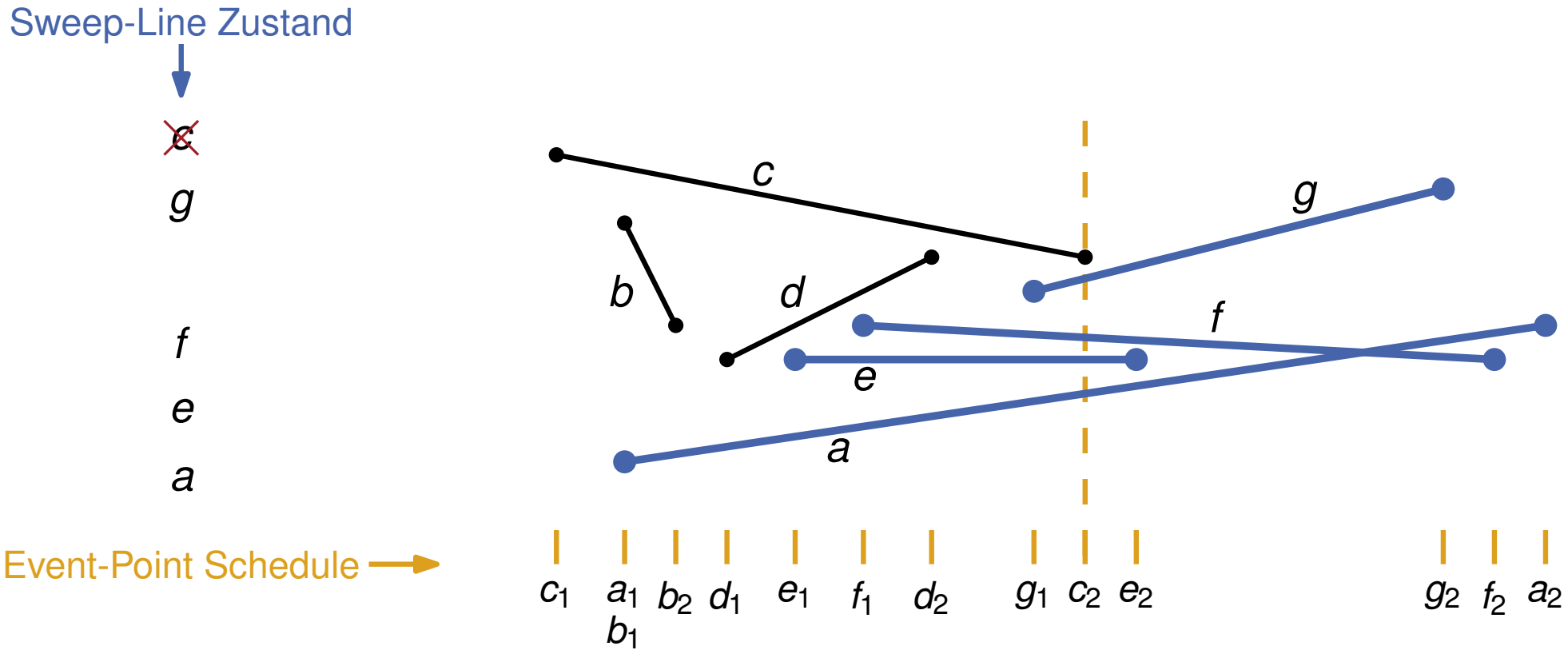
Sweep-Line Algorithmus – Beispiel



g schneidet weder *f* noch *c*

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

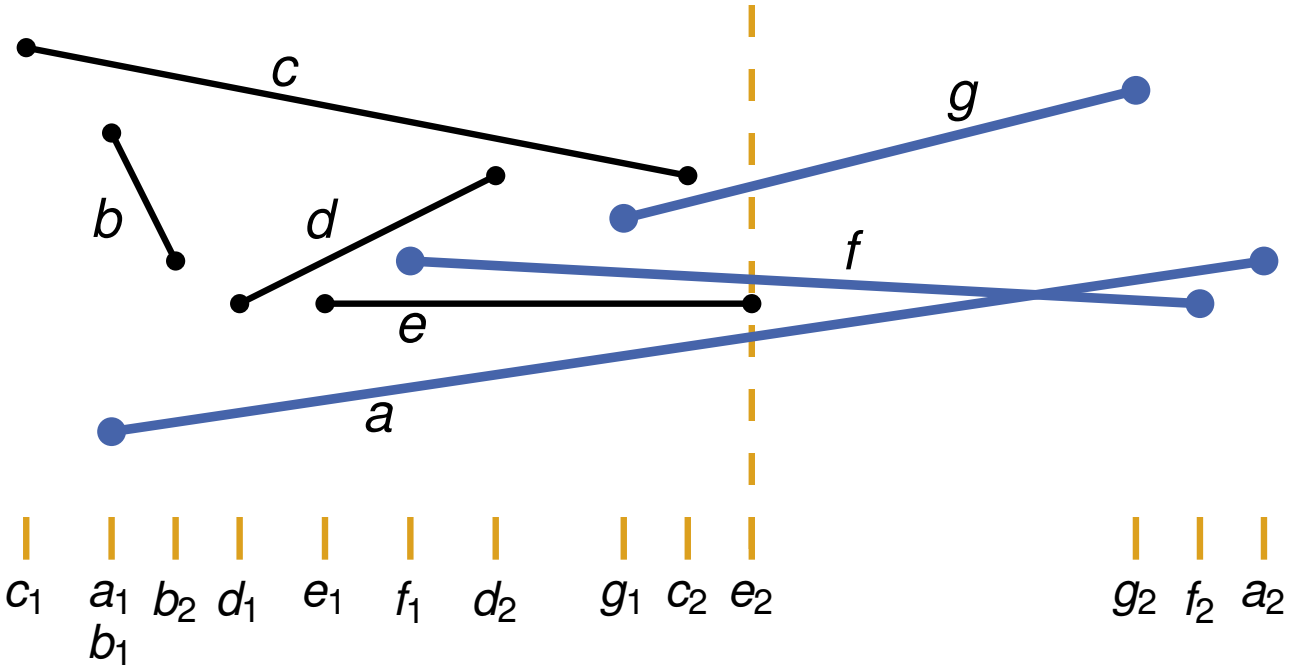
Sweep-Line Algorithmus – Beispiel



Sweep-Line Algorithmus – Beispiel

Sweep-Line Zustand

Event-Point Schedule



f schneidet a! $\Rightarrow$ gib TRUE zurück

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmische Geometrie

Grundlegendes

- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmisches

- Was ist die Grundidee eines Sweep-Line Algorithmus? Was speichert man im Laufe des Algorithmus?
- Wie und mit welcher Laufzeit kann man Testen, ob es unter einer Menge von Strecken ein Paar gibt, das sich schneidet?

Algorithmische Geometrie

Grundlegendes

- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmisches

- Was ist die Grundidee eines Sweep-Line Algorithmus? Was speichert man im Laufe des Algorithmus?
- Wie und mit welcher Laufzeit kann man Testen, ob es unter einer Menge von Strecken ein Paar gibt, das sich schneidet?
- Wie funktionieren der Graham Scan bzw. der Jarvis' March (gift wrapping)? Welche Laufzeit haben sie?

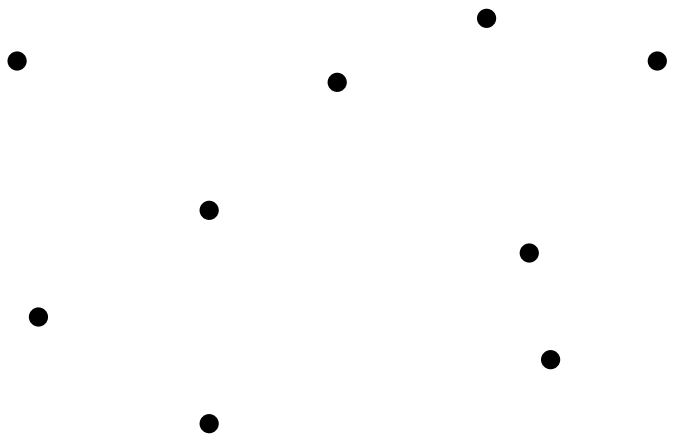
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

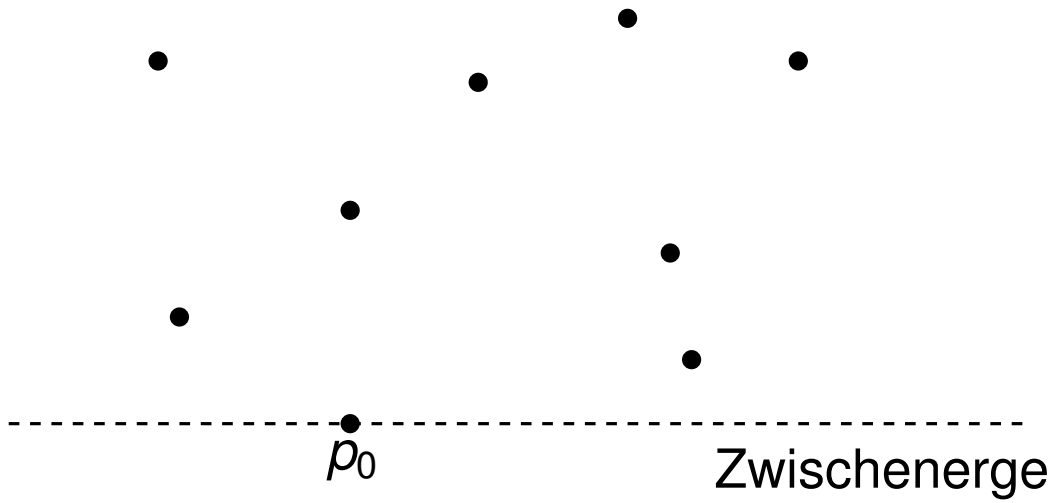
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle: p_0

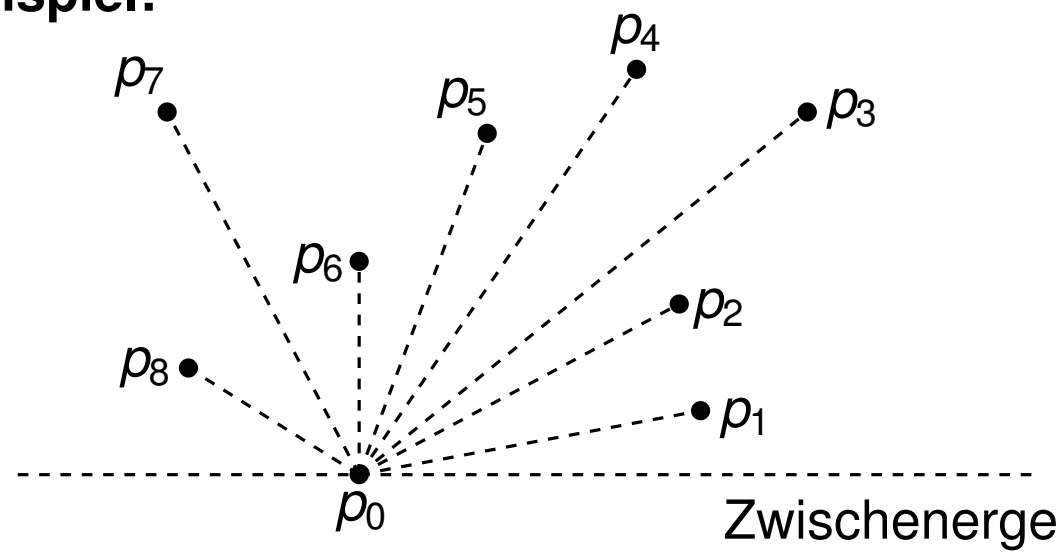
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle: p_0

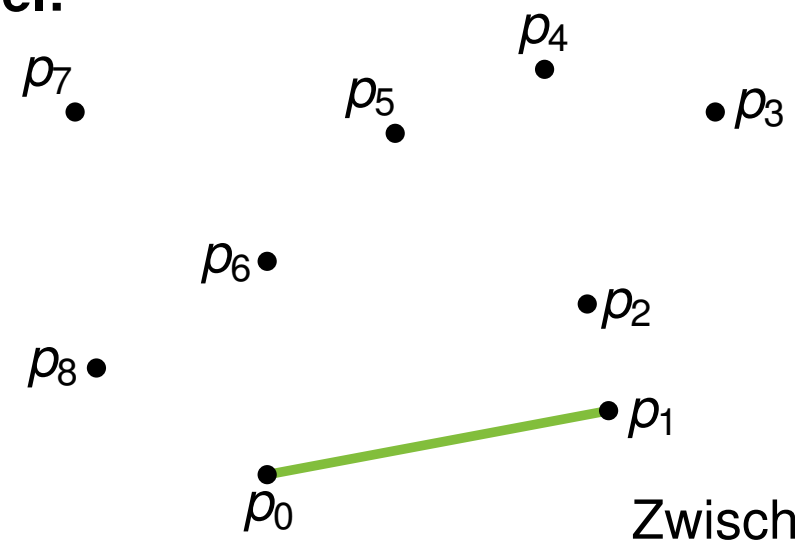
Der Graham Scan (1972)

Idee:

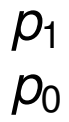
1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:



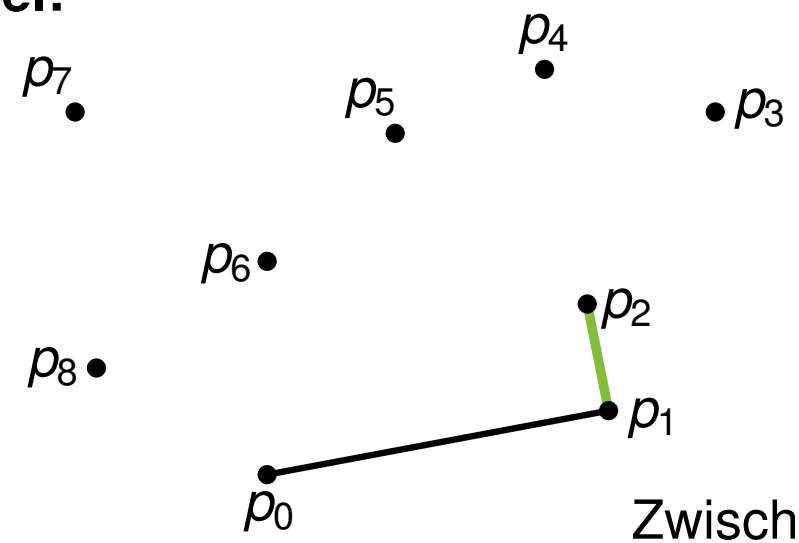
Der Graham Scan (1972)

Idee:

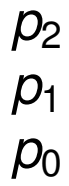
1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:



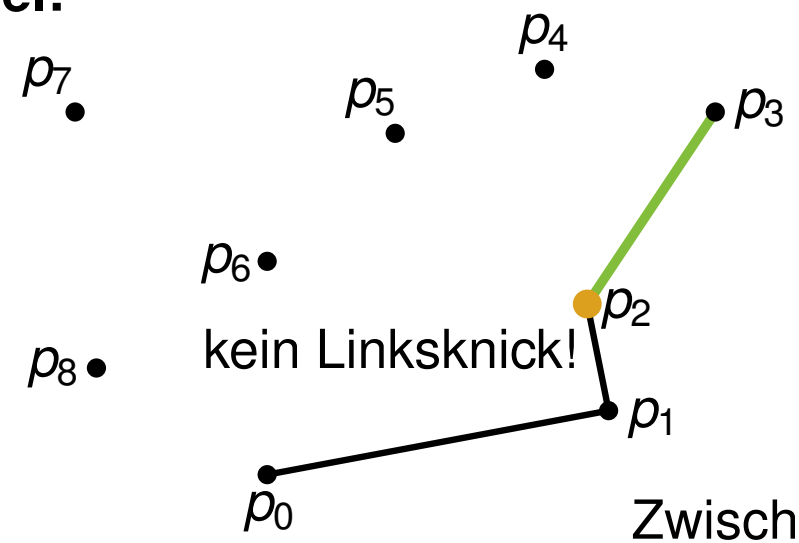
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_3
p_2
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

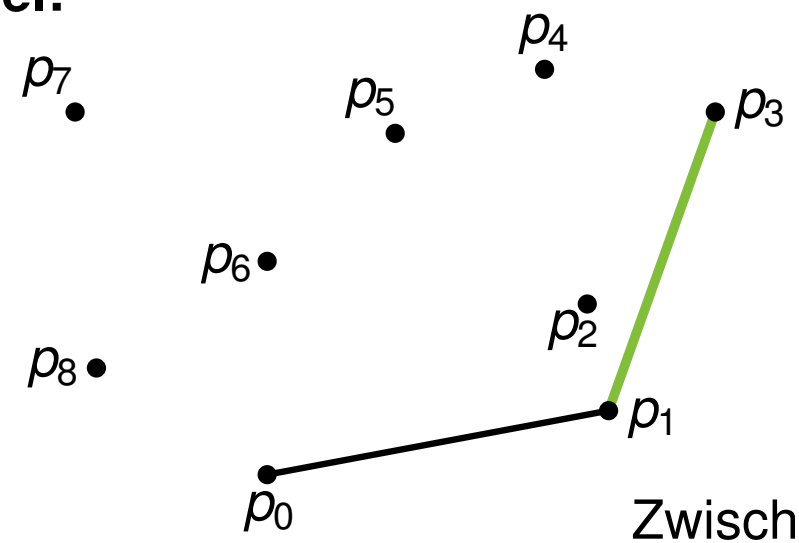
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_3
p_1
p_0

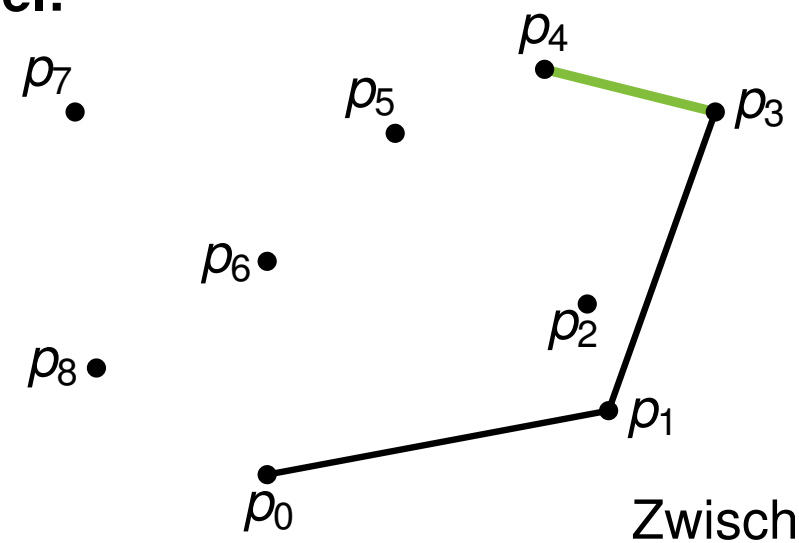
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_4
p_3
p_1
p_0

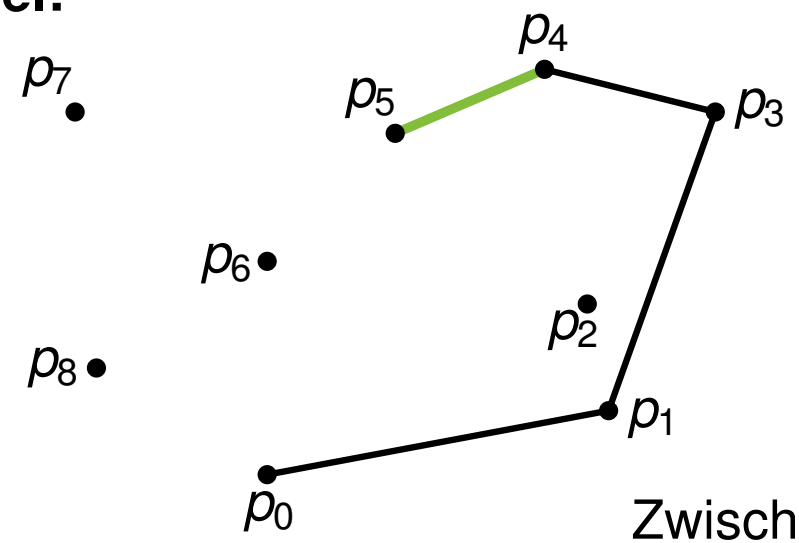
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_5
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

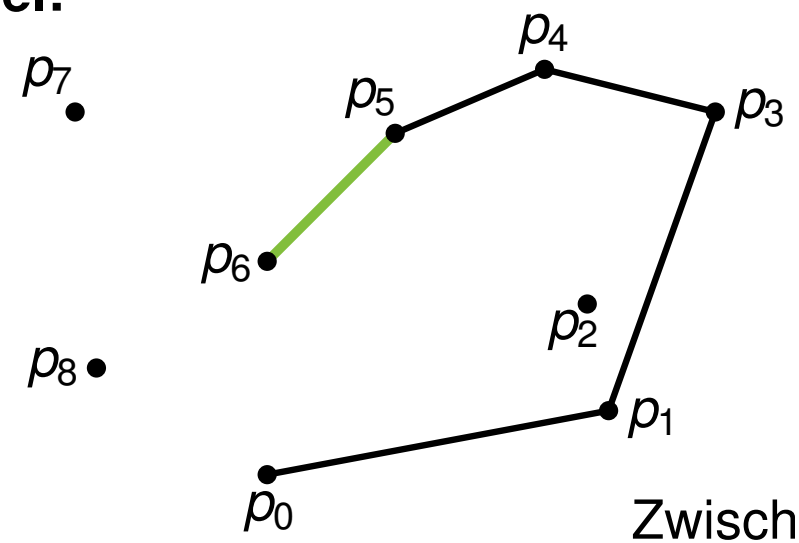
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_6
p_5
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

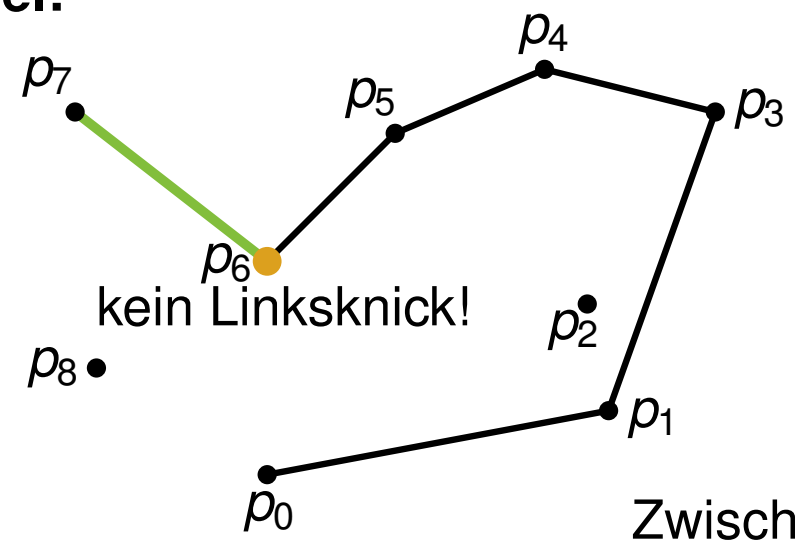
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_7
p_6
p_5
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

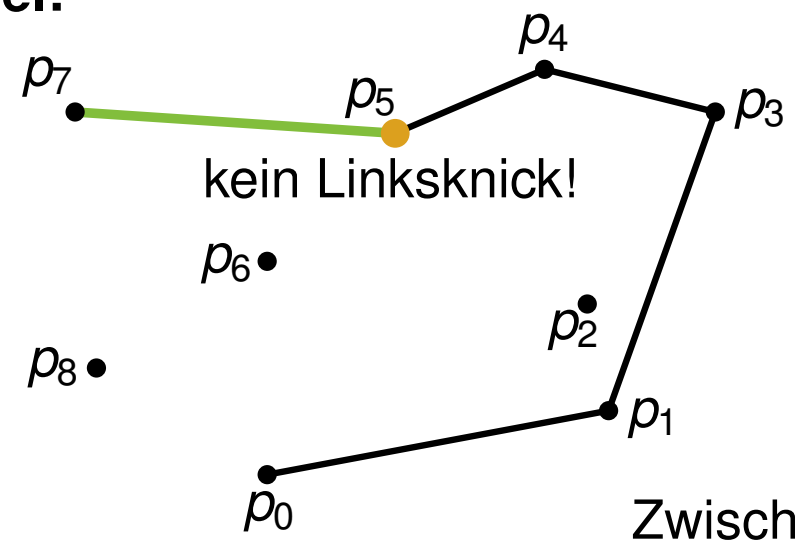
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_7
p_5
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

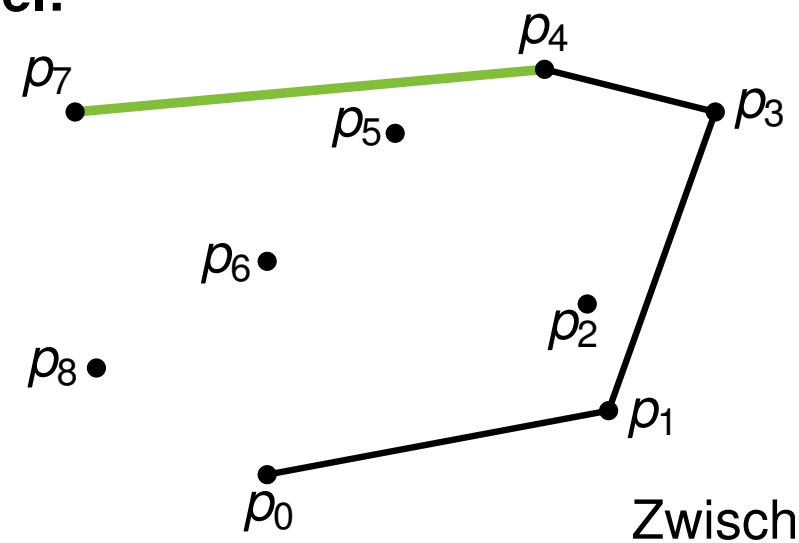
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_7
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

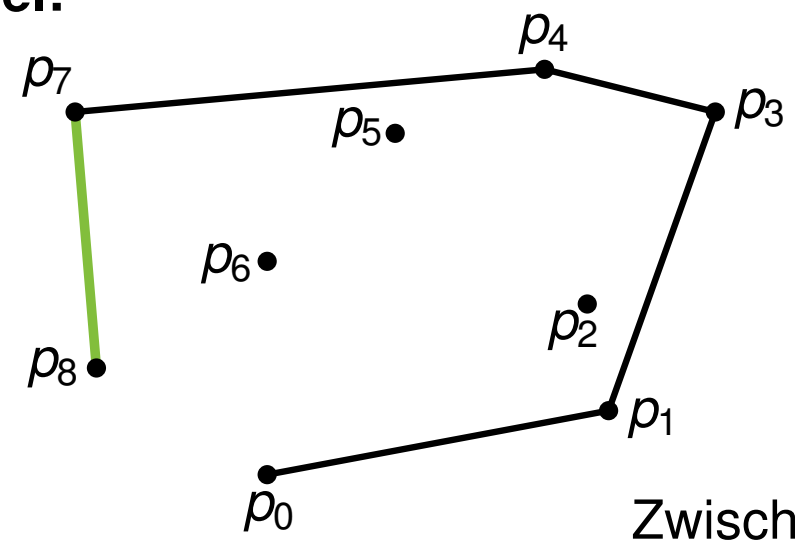
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_8
p_7
p_4
p_3
p_1
p_0

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

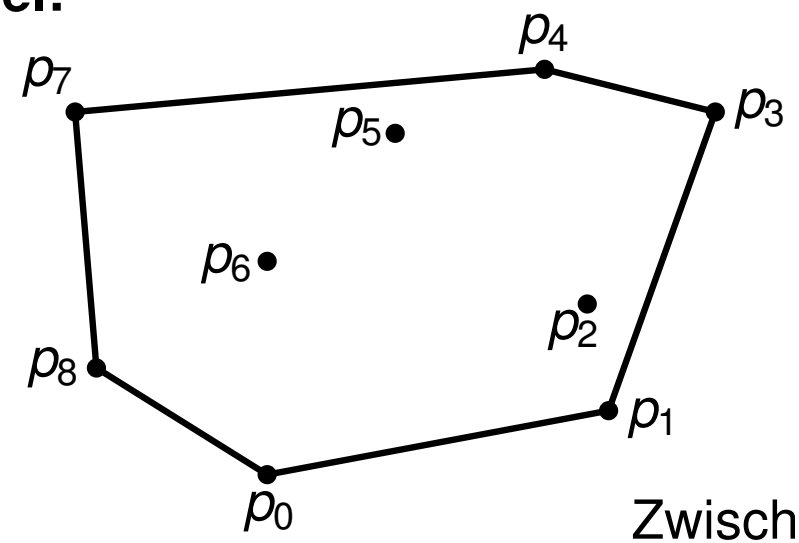
Der Graham Scan (1972)

Idee:

1. Bestimme untersten Punkte p_0 (falls nicht eindeutig, dann den Linkesten).
2. Ordne die restlichen Punkte p_i nach dem Winkel zwischen der Horizontalen durch p_0 und $\overline{p_0 p_i}$.
3. Durchlaufe die Punkte $p_0, p_1, \ldots, p_{n-1}$ in dieser Reihenfolge und konstruiere sukzessive die konvexe Hülle $H(Q)$ nach folgender Regel:

   Wird beim Durchlaufen der Strecken $\overline{p_i p_j}$ und $\overline{p_j p_k}$ nicht links abgebogen, so ist p_j nicht aus dem Rand von $H(Q)$.

Beispiel:



Zwischenergebnis für die konvexe Hülle:

p_8
p_7
p_4
p_3
p_1
p_0

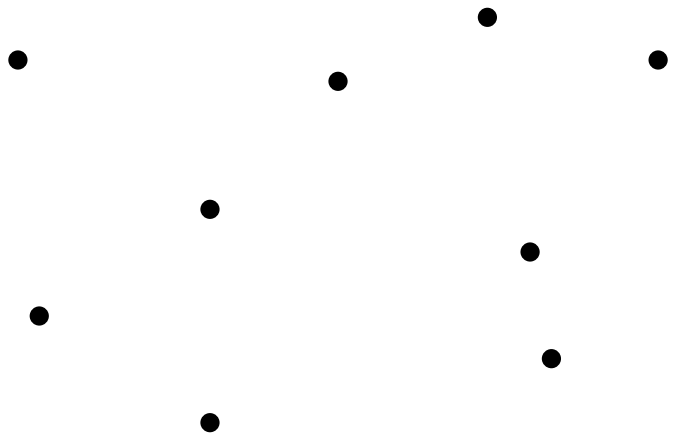
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

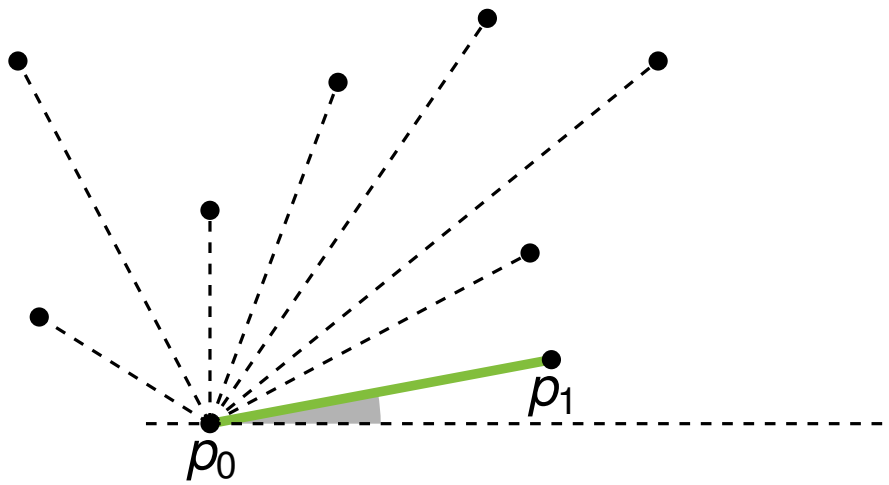


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

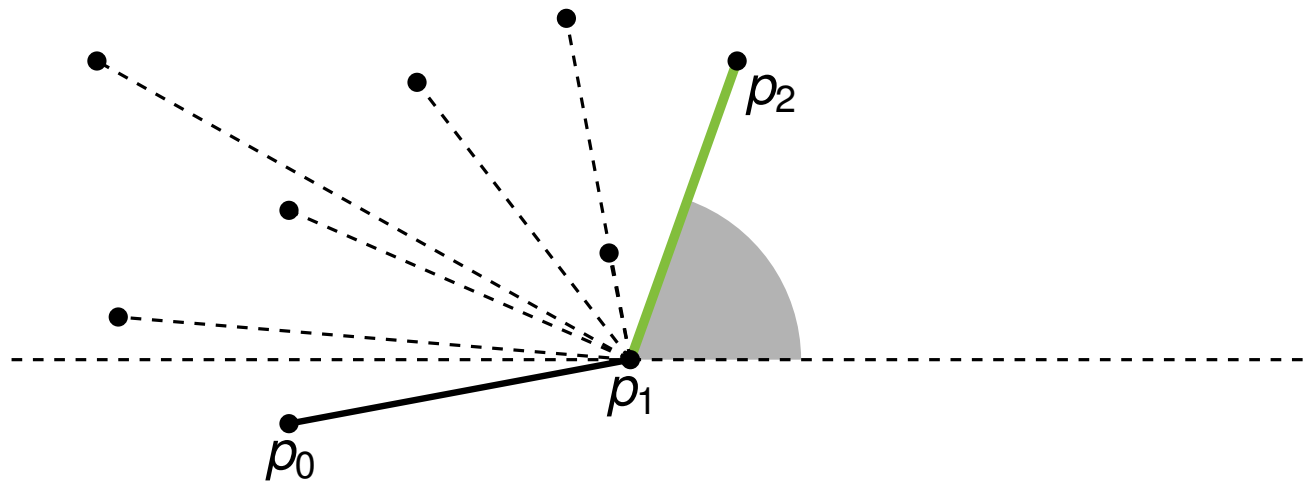


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:



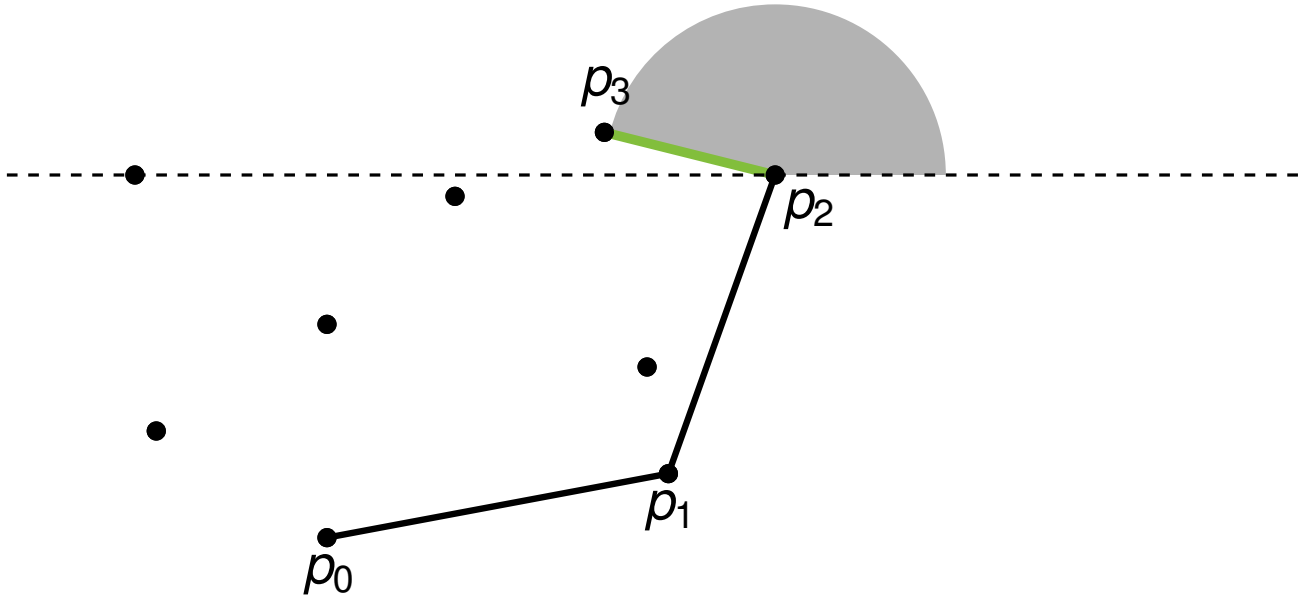
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

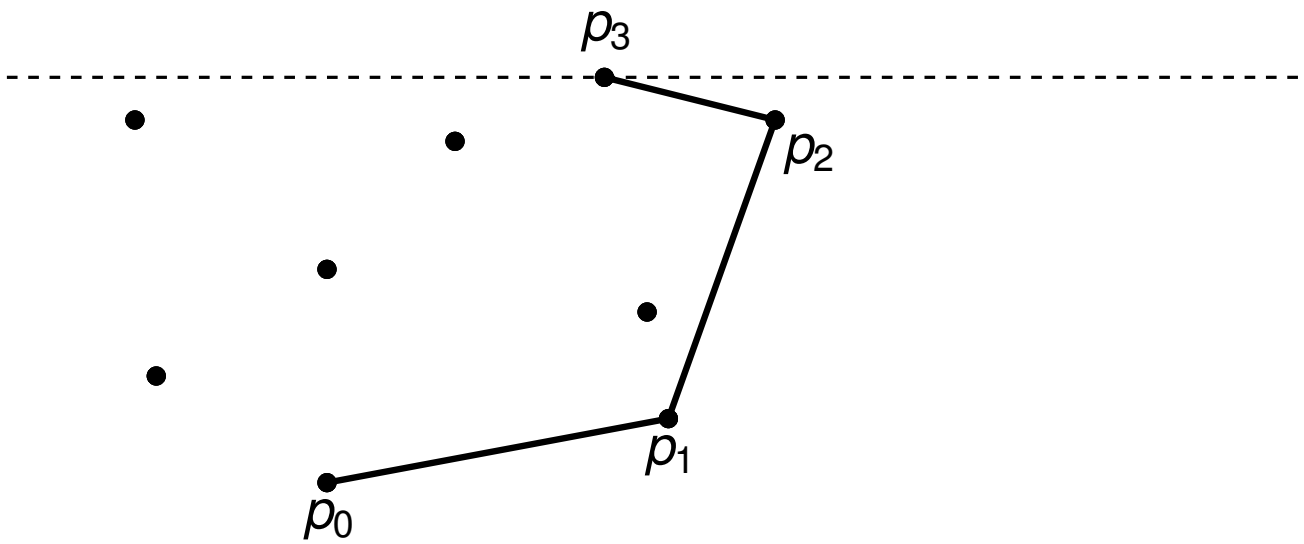


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

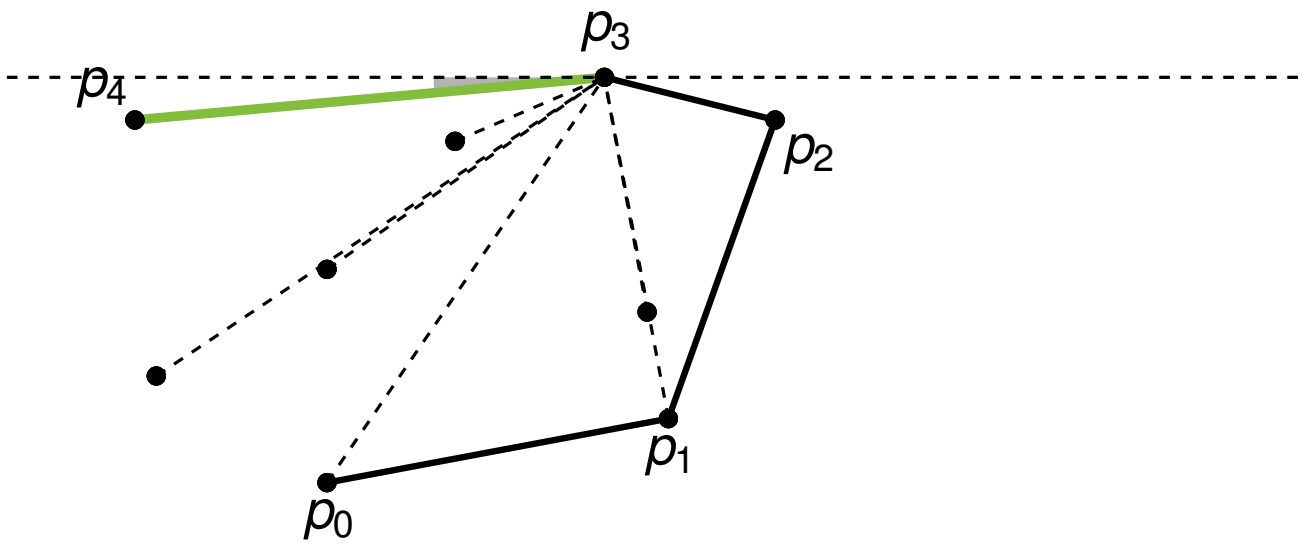


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

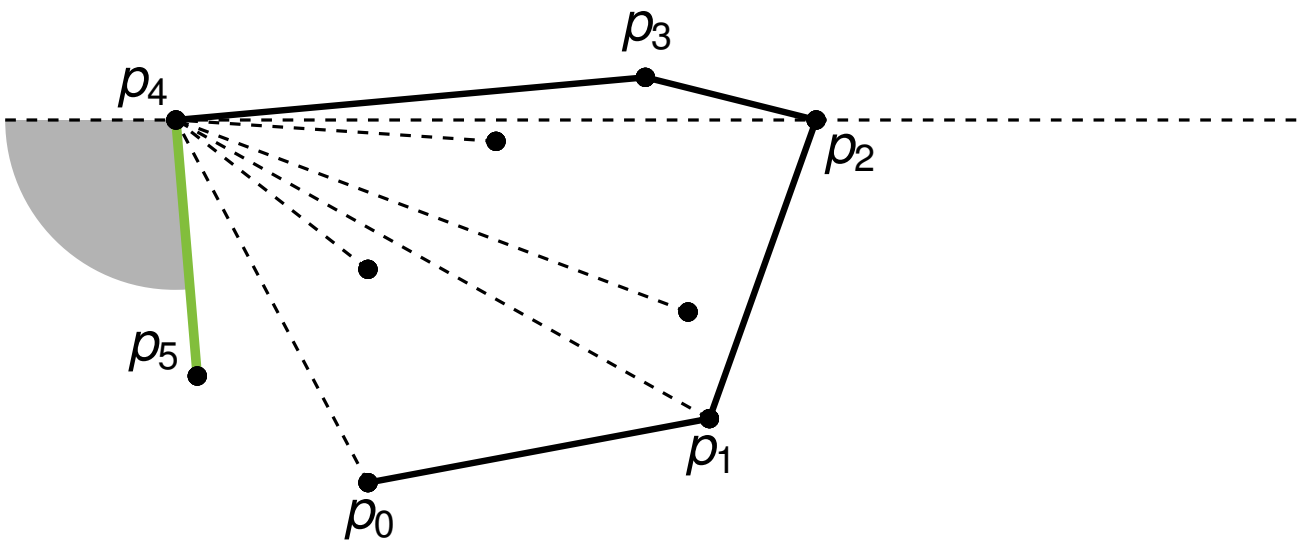


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:

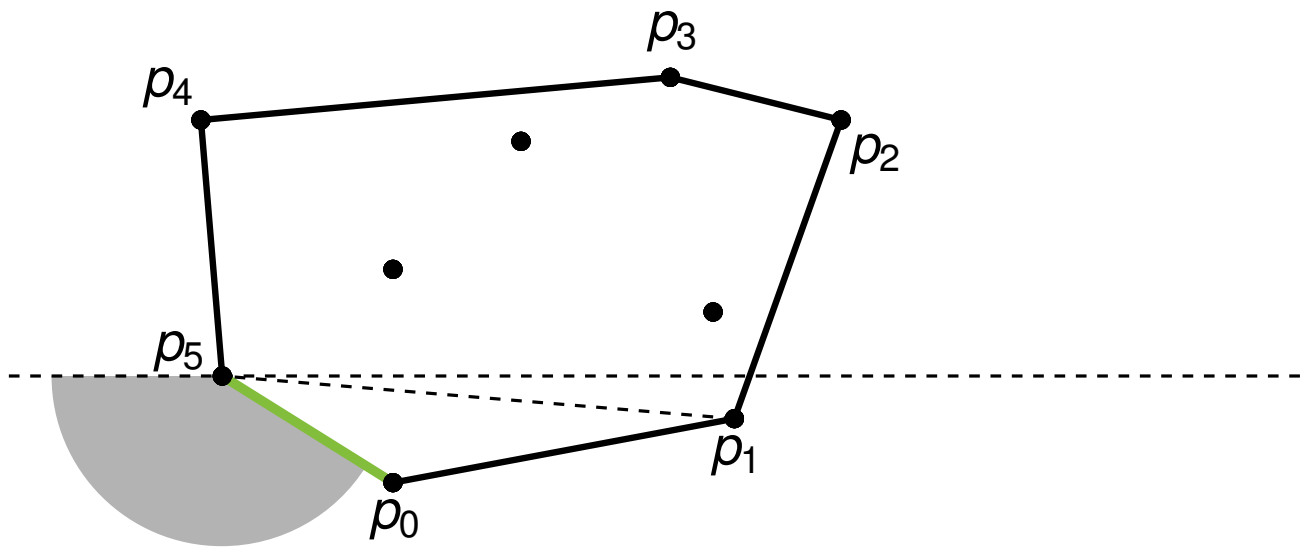


Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:



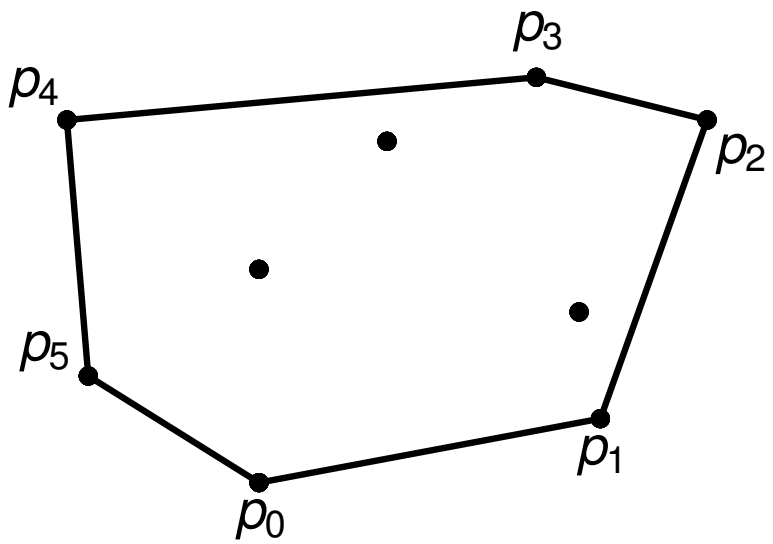
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Gift Wrapping Algorithmus (Jarvis' March)

Idee:

1. Berechne zwei Teile der konvexen Hülle getrennt: Pfad vom untersten Punkt p_0 der Konvexen Hülle zum Obersten („Rechtskette“) und dann Pfad vom Obersten zum Untersten („Linkskette“).
2. Berechnung der „Rechtskette“: Wähle p_{i+1} sodass der Winkel zwischen der Horizontalen (nach rechts) durch p_i und der Strecke $\overline{p_i p_{i+1}}$ minimal ist.
3. Berechnung der „Linkskette“: Analog von oben nach unten.

Beispiel:



Algorithmische Geometrie

Grundlegendes

- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmisches

- Was ist die Grundidee eines Sweep-Line Algorithmus? Was speichert man im Laufe des Algorithmus?
- Wie und mit welcher Laufzeit kann man Testen, ob es unter einer Menge von Strecken ein Paar gibt, das sich schneidet?
- Wie funktionieren der Graham Scan bzw. der Jarvis' March (gift wrapping)? Welche Laufzeit haben sie?

Algorithmische Geometrie

Karlsruher Institut für Technologie

Grundlegendes

- Was bedeuten die Begriffe Strecke, (einfaches/konvexes) Polygon, Konvexkombination konvex, konvexe Hülle, umschließendes Rechteck?
- Wie testet man, auf welcher Seite einer Gerade ein Punkt liegt?
- Wie testet man, ob zwei Strecken sich schneiden?

Algorithmisches

- Was ist die Grundidee eines Sweep-Line Algorithmus? Was speichert man im Laufe des Algorithmus?
- Wie und mit welcher Laufzeit kann man Testen, ob es unter einer Menge von Strecken ein Paar gibt, das sich schneidet?
- Wie funktionieren der Graham Scan bzw. der Jarvis' March (gift wrapping)? Welche Laufzeit haben sie?
- Warum liegt die Berechnung einer konvexen Hülle in $\Omega(n \log n)$?

String-Matching

Grundlegendes

- Was ist die Problemstellung?
- Warum ist $\Omega(n + m)$ eine untere Schranke?
- Wie hängt die Wahl des Algorithmus von der Art der Anfrage ab?

String-Matching

Grundlegendes

- Was ist die Problemstellung?
- Warum ist $\Omega(n+m)$ eine untere Schranke?
- Wie hängt die Wahl des Algorithmus von der Art der Anfrage ab?

Algorithmisches

- Was ist die Idee bei Rabin & Karp?

String-Matching

Grundlegendes

- Was ist die Problemstellung?
- Warum ist $\Omega(n+m)$ eine untere Schranke?
- Wie hängt die Wahl des Algorithmus von der Art der Anfrage ab?

Algorithmisches

- Was ist die Idee bei Rabin & Karp?
- Wie funktioniert String-Matching mit endlichen Automaten? Laufzeit?

String-Matching

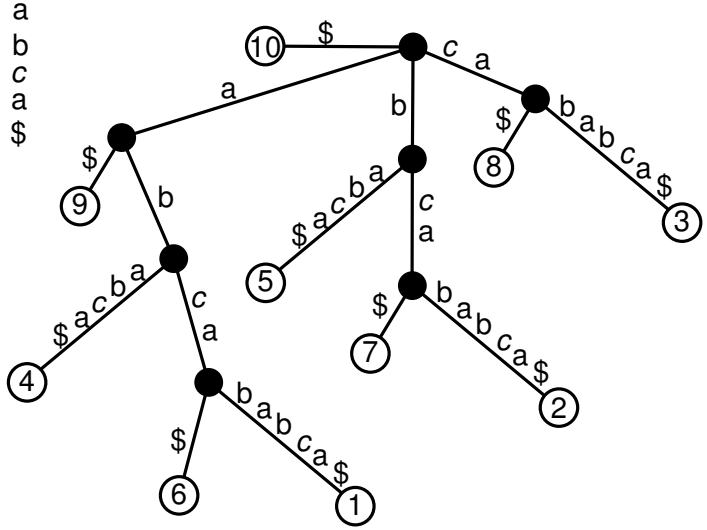
Grundlegendes

- Was ist die Problemstellung?
- Warum ist $\Omega(n+m)$ eine untere Schranke?
- Wie hängt die Wahl des Algorithmus von der Art der Anfrage ab?

Beispiel: T= a b c a b a b c a $

$A =$

10	9	4	6	1	5	7	2	8	3
$	a $	a b a b c a $	a b c a $	a b c a b a b c a $	b a b c a $	b c a $	b c a b a b c a $	c a $	c a b a b c a $



Algorithmisches

- Was ist die Idee bei Rabin & Karp?
- Wie funktioniert String-Matching mit endlichen Automaten? Laufzeit?
- Was ist ein Suffix-Baum? Was ist ein Suffix-Array?
- Wie kann man mit diesen Datenstrukturen Muster finden? Laufzeit?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

String-Matching

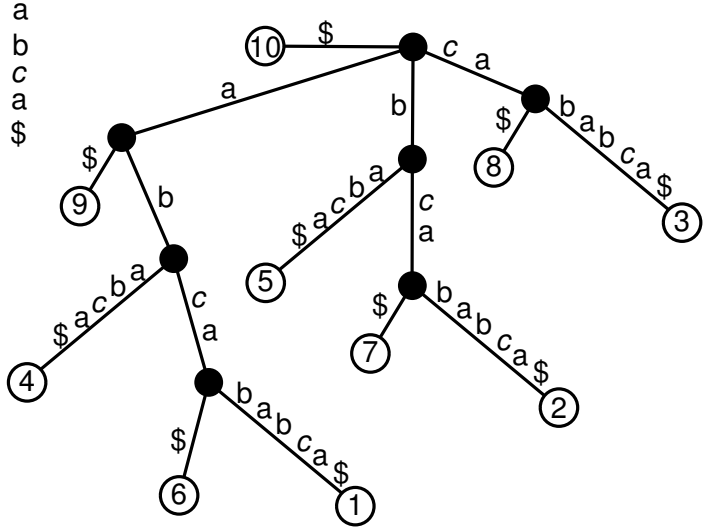
Grundlegendes

- Was ist die Problemstellung?
- Warum ist $\Omega(n+m)$ eine untere Schranke?
- Wie hängt die Wahl des Algorithmus von der Art der Anfrage ab?

Beispiel: T= a b c a b a b c a $

$A =$

10	9	4	6	1	5	7	2	8	3
$	a $	a b a b c a $	a b c a $	a b c a b a b c a $	b a b c a $	b c a $	b c a b a b c a $	c a $	c a b a b c a $



Algorithmisches

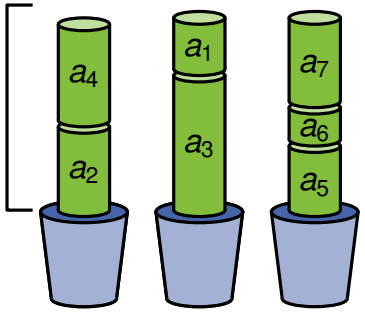
- Was ist die Idee bei Rabin & Karp?
- Wie funktioniert String-Matching mit endlichen Automaten? Laufzeit?
- Was ist ein Suffix-Baum? Was ist ein Suffix-Array?
- Wie kann man mit diesen Datenstrukturen Muster finden? Laufzeit?
- In welcher Laufzeit können Suffix-Bäume und Suffix-Arrays konstruiert werden?
- Wie kann man sie konstruieren (nicht die $O(n)$ Konstruktion)?

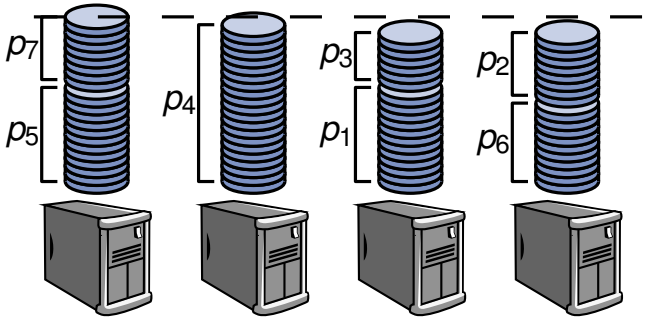
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Approximierende Algorithmen

Grundlegendes

- Was sind Approximationsalgorithmen mit relativer Gütegarantie?[1]
- Was sind PAS, FPAS, APAS, AFPAS?
- Wie sind Multiprozessor-Scheduling und Bin Packing definiert?



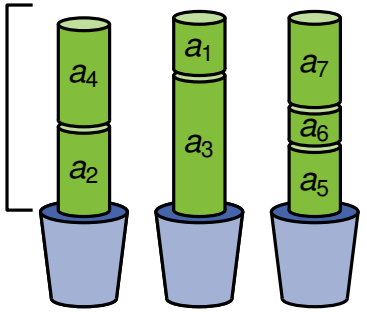


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Approximierende Algorithmen

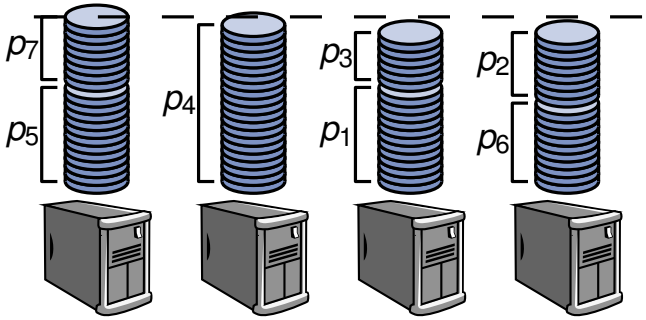
Grundlegendes

- Was sind Approximationsalgorithmen mit relativer Gütegarantie?[1]
- Was sind PAS, FPAS, APAS, AFPAS?
- Wie sind Multiprozessor-Scheduling und Bin Packing definiert?

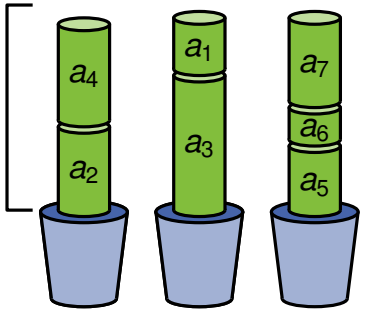


Algorithmisches

- Wie funktioniert LIST SCHEDULING und was ist die Approximationsgüte? Wie kann man damit zu einem PAS kommen?



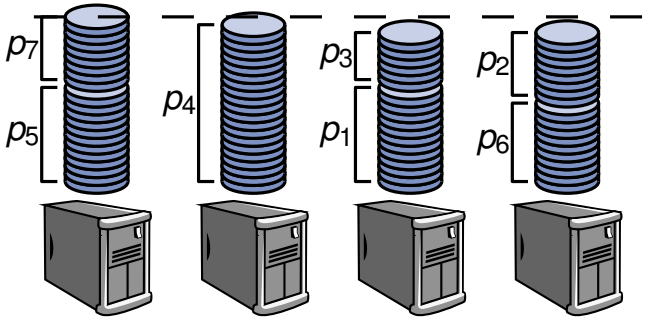
Approximierende Algorithmen



Grundlegendes

- Was sind Approximationsalgorithmen mit relativer Gütegarantie?[1]
- Was sind PAS, FPAS, APAS, AFPAS?
- Wie sind Multiprozessor-Scheduling und Bin Packing definiert?

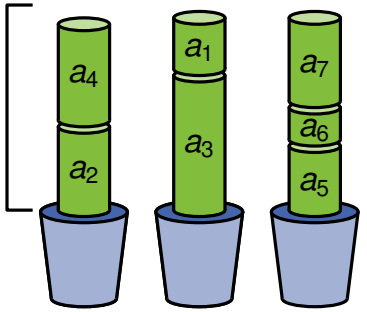
Algorithmisches



- Wie funktioniert LIST SCHEDULING und was ist die Approximationsgüte? Wie kann man damit zu einem PAS kommen?
- Was ist NEXT FIT (Bin Packing)? Ist die Approximationsrate von 2 scharf?
- Was ist FIRST FIT? Wie ist die (asymptotische) Approximationsrate hier?

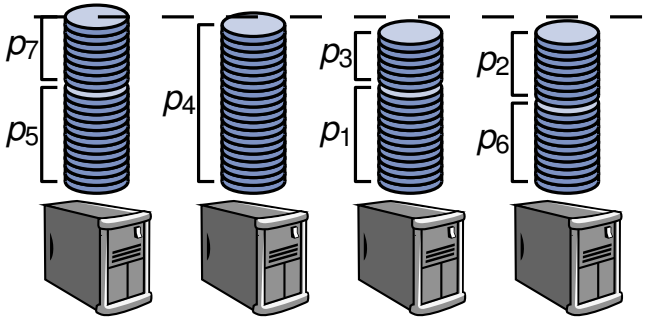
Approximierende Algorithmen

Grundlegendes



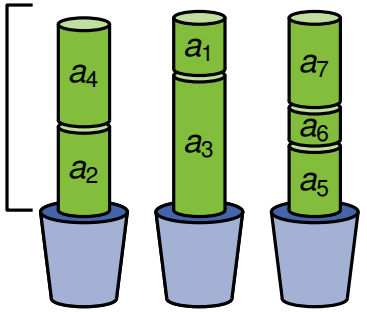
- Was sind Approximationsalgorithmen mit relativer Gütegarantie?[1]
- Was sind PAS, FPAS, APAS, AFPAS?
- Wie sind Multiprozessor-Scheduling und Bin Packing definiert?

Algorithmisches



- Wie funktioniert LIST SCHEDULING und was ist die Approximationsgüte? Wie kann man damit zu einem PAS kommen?
- Was ist NEXT FIT (Bin Packing)? Ist die Approximationsrate von 2 scharf?
- Was ist FIRST FIT? Wie ist die (asymptotische) Approximationsrate hier?
- Was ist RESTRICTED BIN PACKING (wenige verschiedene Elementtypen + Mindestgröße) und wie kann es als ILP formuliert werden?

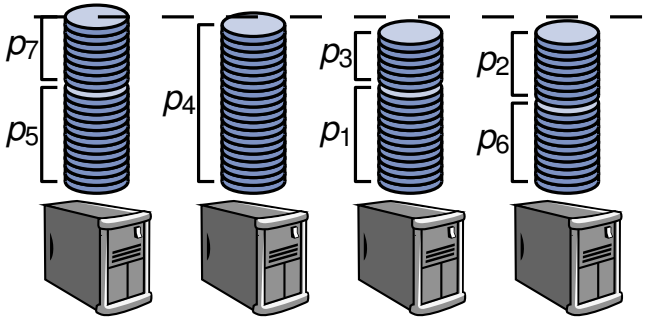
Approximierende Algorithmen



Grundlegendes

- Was sind Approximationsalgorithmen mit relativer Gütegarantie?[1]
- Was sind PAS, FPAS, APAS, AFPAS?
- Wie sind Multiprozessor-Scheduling und Bin Packing definiert?

Algorithmisches

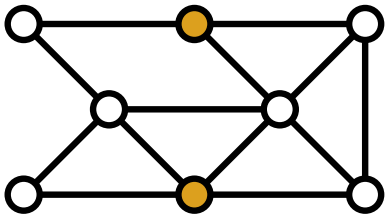


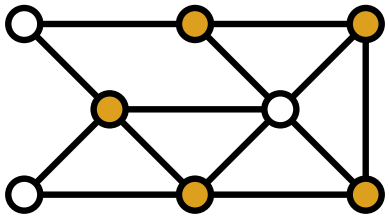
- Wie funktioniert LIST SCHEDULING und was ist die Approximationsgüte? Wie kann man damit zu einem PAS kommen?
- Was ist NEXT FIT (Bin Packing)? Ist die Approximationsrate von 2 scharf?
- Was ist FIRST FIT? Wie ist die (asymptotische) Approximationsrate hier?
- Was ist RESTRICTED BIN PACKING (wenige verschiedene Elementtypen + Mindestgröße) und wie kann es als ILP formuliert werden?
- Wie groß ist das ILP? Hängt die Größe noch von n ab?
- Wie wird man bei einer allgemeinen BIN PACKING Instanz die kleinen Elemente los und was ist lineares gruppieren?
- Warum liefert das ein APAS (AFPAS)?

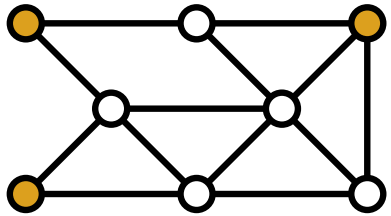
Parametrisierte Algorithmen

Grundlegendes

- Wie sind die Probleme VERTEX COVER, INDEPENDENT SET und DOMINATING SET definiert?
- Was ist ein FPT-Algorithmus?



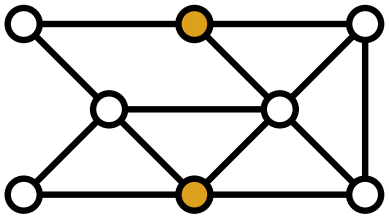
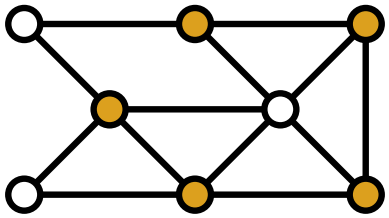
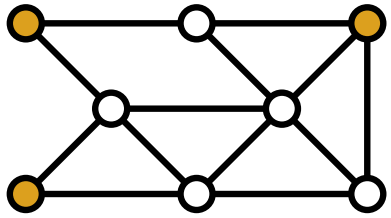




Parametrisierte Algorithmen

Grundlegendes

- Wie sind die Probleme VERTEX COVER, INDEPENDENT SET und DOMINATING SET definiert?
- Was ist ein FPT-Algorithmus?

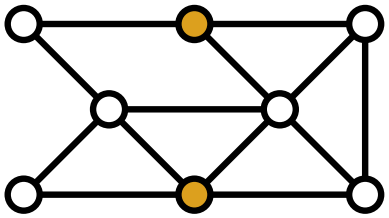
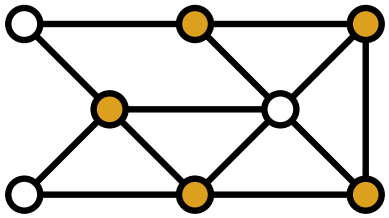
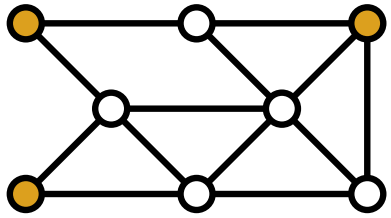
Algorithmisches

- Was ist Kernbildung? Wie funktionieren Tiefenbeschränkte Suchbäume?
- Wie funktioniert die Kernbildung bei VERTEX COVER? Laufzeit?

Parametrisierte Algorithmen

Grundlegendes

- Wie sind die Probleme VERTEX COVER, INDEPENDENT SET und DOMINATING SET definiert?
- Was ist ein FPT-Algorithmus?

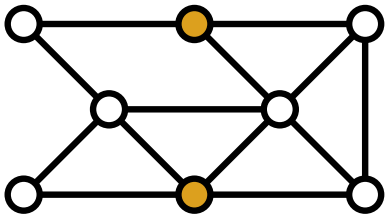
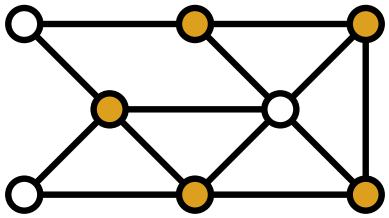
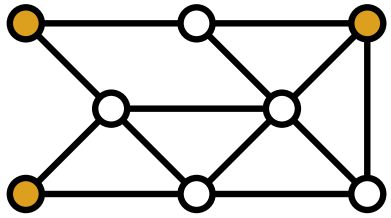
Algorithmisches

- Was ist Kernbildung? Wie funktionieren Tiefenbeschränkte Suchbäume?
- Wie funktioniert die Kernbildung bei VERTEX COVER? Laufzeit?
- Wie funktioniert der Entscheidungsbaum für VERTEX COVER mit Verzweigungsvektor $(1, 1)$? Laufzeit?
- Idee, wie die Laufzeit verbessert werden kann?

Parametrisierte Algorithmen

Grundlegendes

- Wie sind die Probleme VERTEX COVER, INDEPENDENT SET und DOMINATING SET definiert?
- Was ist ein FPT-Algorithmus?

Algorithmisches

- Was ist Kernbildung? Wie funktionieren Tiefenbeschränkte Suchbäume?
- Wie funktioniert die Kernbildung bei VERTEX COVER? Laufzeit?
- Wie funktioniert der Entscheidungsbaum für VERTEX COVER mit Verzweigungsvektor $(1, 1)$? Laufzeit?
- Idee, wie die Laufzeit verbessert werden kann?
- Wie kann VERTEX COVER als ILP formuliert werden? Wie hilft diese ILP, wenn man einen FPT-Algorithmus sucht?

Online Algorithmen

Grundlegendes

- Was ist ein Online-Algorithmus? Wann ist ein Algorithmus kompetitiv?
- Was hat das mit Approximationsalgorithmen zu tun?

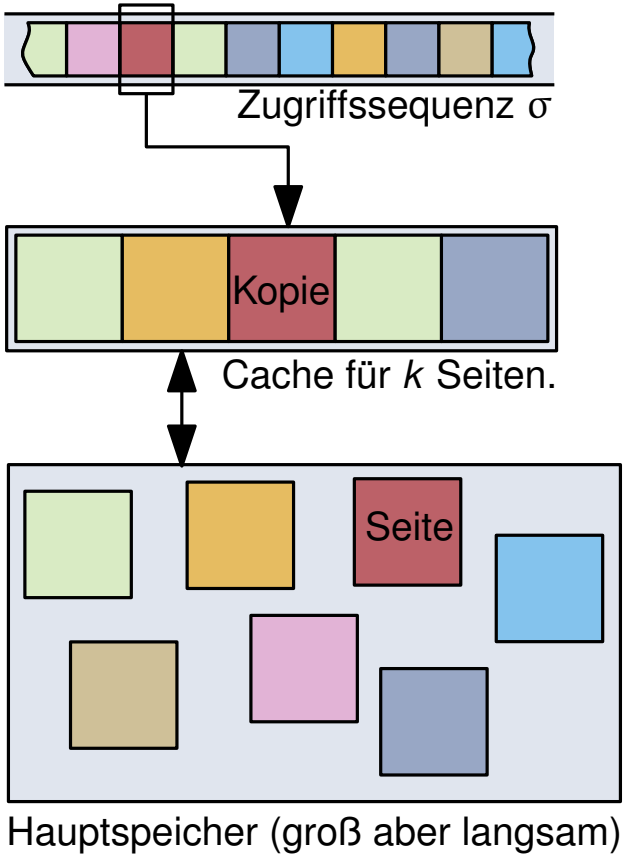
Online Algorithmen

Grundlegendes

- Was ist ein Online-Algorithmus? Wann ist ein Algorithmus kompetitiv?
- Was hat das mit Approximationsalgorithmen zu tun?

Paging

- Was ist die Problemstellung? Welche Strategien gibt es?



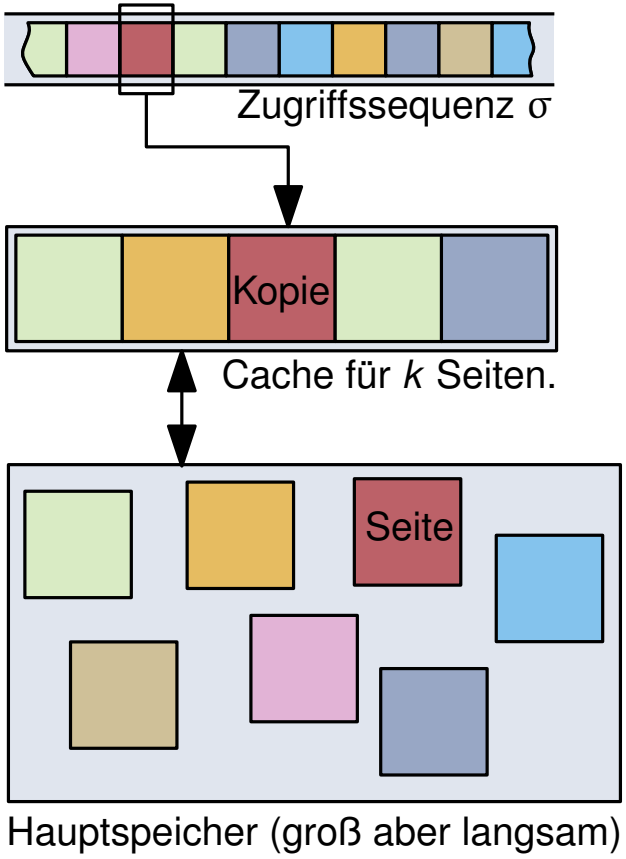
Online Algorithmen

Grundlegendes

- Was ist ein Online-Algorithmus? Wann ist ein Algorithmus kompetitiv?
- Was hat das mit Approximationsalgorithmen zu tun?

Paging

- Was ist die Problemstellung? Welche Strategien gibt es?
- Warum vergleicht man sie mit offline Algorithmus?



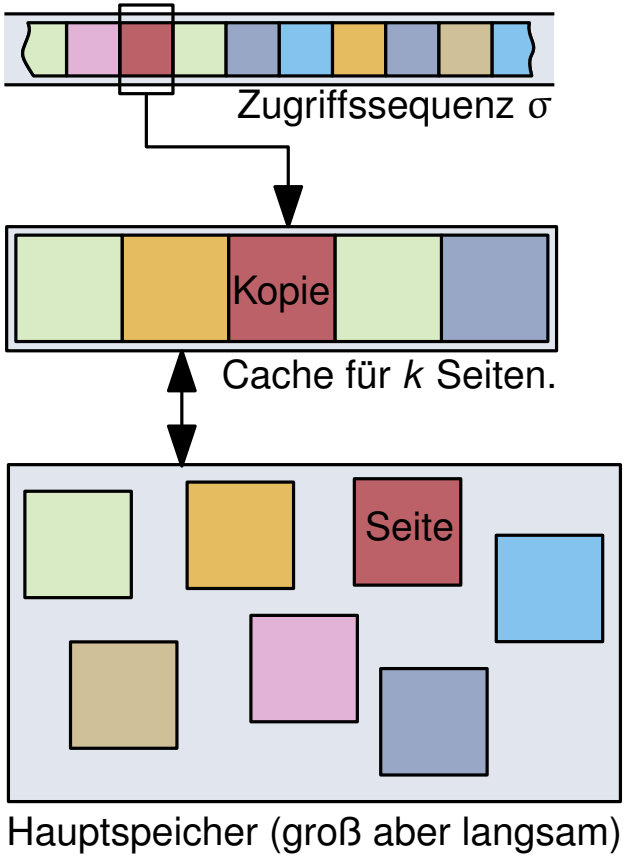
Online Algorithmen

Grundlegendes

- Was ist ein Online-Algorithmus? Wann ist ein Algorithmus kompetitiv?
- Was hat das mit Approximationsalgorithmen zu tun?

Paging

- Was ist die Problemstellung? Welche Strategien gibt es?
- Warum vergleicht man sie mit offline Algorithmus?
- Warum kann man keine Gütegarantie besser als k erreichen?
- Was ist (h, k)-Paging? Wann ist ein Algorithmus konservativ?
- Inwiefern hilft (h, k)-Paging bei der Analyse?



Parallele Algorithmen

Grundlegendes

- Wie misst man die Qualität eines parallelen Algorithmus?
- Was ist der speed-up und wann ist ein Algorithmus kostenoptimal?

Parallele Algorithmen

Grundlegendes

- Wie misst man die Qualität eines parallelen Algorithmus?
- Was ist der speed-up und wann ist ein Algorithmus kostenoptimal?

Algorithmisches

- Wie kann man Summen (und andere binäre Operationen) parallel berechnen?

Parallele Algorithmen

Grundlegendes

- Wie misst man die Qualität eines parallelen Algorithmus?
- Was ist der speed-up und wann ist ein Algorithmus kostenoptimal?

Algorithmisches

- Wie kann man Summen (und andere binäre Operationen) parallel berechnen?
- Was sind Präfixsummen und wie kann man sie parallel berechnen?

Parallele Algorithmen

Grundlegendes

- Wie misst man die Qualität eines parallelen Algorithmus?
- Was ist der speed-up und wann ist ein Algorithmus kostenoptimal?

Algorithmisches

- Wie kann man Summen (und andere binäre Operationen) parallel berechnen?
- Was sind Präfixsummen und wie kann man sie parallel berechnen?
- Was ist List Ranking?

Parallele Algorithmen

Grundlegendes

- Wie misst man die Qualität eines parallelen Algorithmus?
- Was ist der speed-up und wann ist ein Algorithmus kostenoptimal?

Algorithmisches

- Wie kann man Summen (und andere binäre Operationen) parallel berechnen?
- Was sind Präfixsummen und wie kann man sie parallel berechnen?
- Was ist List Ranking?
- Was ist die Idee zur parallelen Berechnung der Zusammenhangskomponenten eines Graphen? Wie kann das auf MST erweitert werden?

Werbung

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

ICPC

Algorithmische Kartografie

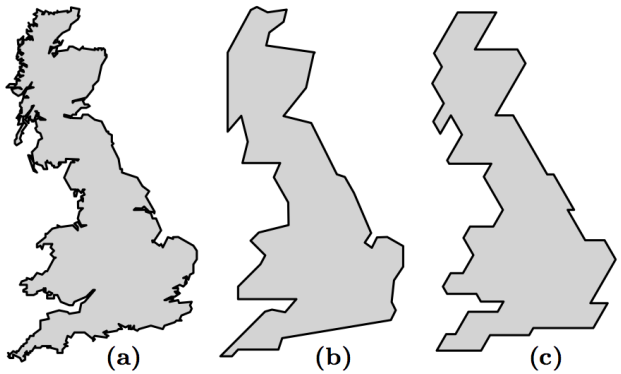
- Mastervorlesung mit Übung 3 SWS, 5LP
- Termine: Di 9:45, Do 9:45
- Dozenten: Martin Nöllenburg, Benjamin Niedermann

Inhalt:

- Algorithmen zur Erstellung von Landkarten und Visualisierungen räumlicher Daten
- geometrische Modellierung kartografischer Probleme
- Beispiele: Vereinfachung und Schematisierung von Polygonen und Kantenzügen, Kartenbeschriftung, Flächenkartogramme, dynamische Landkarten

Anforderungen:

- Solide Grundkenntnisse in Algorithmik
- Interesse an geometrischen Algorithmen

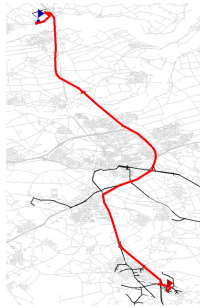


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Werbung SS 2013

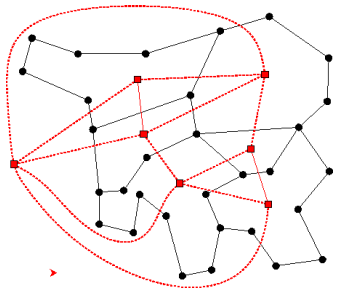
Karlsruher Institut für Technologie

Lehrveranstaltungen nächstes Semester



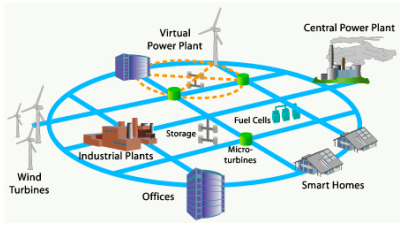
- Routenplanung (Master)
  - Montag 14:00 – 15:30
  - Mittwoch 11:30 – 13:00

- Algorithmische Kartografie (Master)
  - Vorlesung: Dienstag, 9:45 – 11:15
  - Übung: Donnerstag, 9:45 – 11:15



- Algorithmen für planare Graphen (Bachelor)
  - Dienstag 14:00 – 15:30
  - Mittwoch 14:00 – 15:30

- Seminar Algorithms for Future Energy Systems (Master)
  - Dienstag: 15:45 – 17:15



i11www.iti.kit.edu/teaching

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner