3. Übung Algorithmentechnik

Lehrstuhl für Algorithmik I
Institut für Theoretische Informatik
Universität Karlsruhe (TH)

04.12.2008

Kreuzende Schnitte

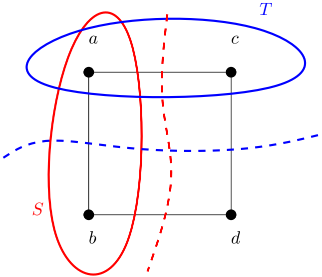
- $G = (V, E)$ Graph
- $(S, V \setminus S)$ und $(T, V \setminus T)$ Schnitte in G
- Schnitte kreuzen sich, wenn

$$
\begin{array}{llll}
A := & S \cap T & \neq \emptyset & \\
B := & S \setminus T & \neq \emptyset & (S \not\subseteq T) \\
C := & T \setminus S & \neq \emptyset & (T \not\subseteq S) \\
D := & V \setminus (S \cup T) & \neq \emptyset & (S \cup T \neq V)
\end{array}
$$

Kreuzende Schnitte

Beispiel

$$
\begin{aligned}
S &:= \{a, b\} \\
T &:= \{a, c\} \\
A &:= S \cap T && = \{a\} \neq \emptyset \\
B &:= S \setminus T && = \{b\} \neq \emptyset \\
C &:= T \setminus S && = \{c\} \neq \emptyset \\
D &:= V \setminus (S \cup T) && = \{d\} \neq \emptyset
\end{aligned}
$$



Kreuzende Schnitte

Bemerkungen

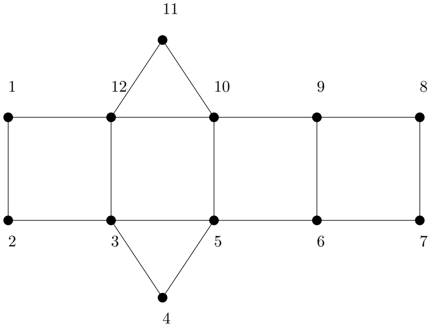
(1) $(S, V \setminus S)$ und $(T, V \setminus T)$ kreuzende Schnitte in G
$\Rightarrow$ $(V \setminus S, S)$ und $(T, V \setminus T)$ kreuzende Schnitte in G (Wohldefiniertheit)

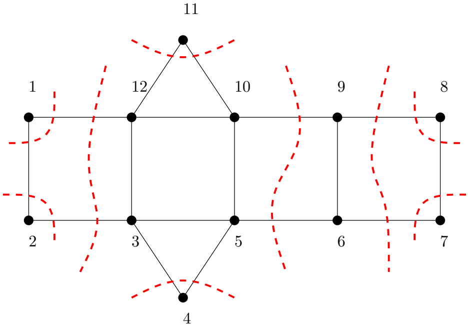
(2) $(S, V \setminus S)$ und $(T, V \setminus T)$ nicht kreuzende Schnitte in G
$\Rightarrow$ dann gilt eine der Eigenschaften

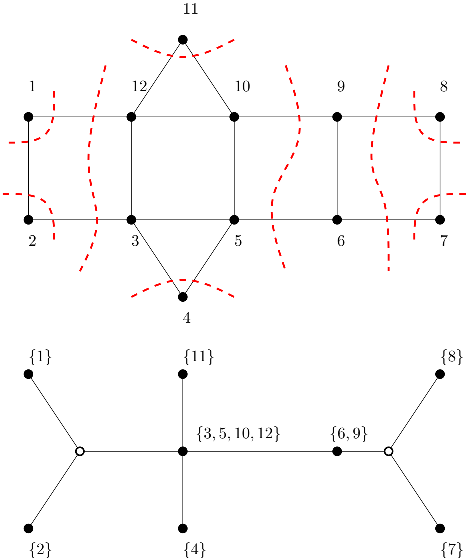
$$S \cap T = \emptyset \tag{1}$$
$$S \cup T = V \tag{2}$$
$$S \subseteq T \tag{3}$$
$$T \subseteq S \tag{4}$$

Repräsentation minimaler Schnitte

- ≫ G Graph, der keine kreuzenden Schnitte minimalen Gewichts hat
- ⇒ Schnitte minimalen Gewichts können als Baum repräsentiert werden
- ≫ jede Kante repräsentiert minimalen Schnitt


11
1
12
10
9
8
2
3
5
6
7
4

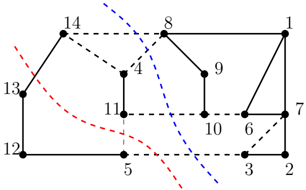

11
1
12
10
9
8
2
3
5
6
7
4

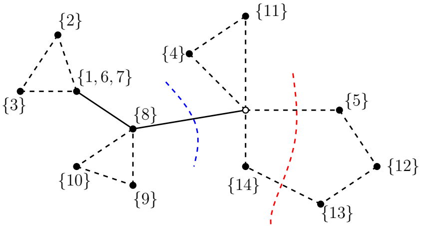

11
1
12
10
9
8
2
3
5
6
7
4
{1}
{11}
{8}
{3, 5, 10, 12}
{6, 9}
{2}
{4}
{7}

Repräsenation minimaler Schnitte

Bemerkungen

- Repräsentation kann für allgemeine Graphen erweitert werden
- Repräsentation ist dann kein Baum, sondern ein sog. Kaktus ($\rightsquigarrow$ Kaktus-Repräsenation)





Aufgabe 1

Aufgabe (Struktur minimaler Schnitte)

Zeigen Sie

- $\gg$ $(S, V \setminus S)$ und $(T, V \setminus T)$ kreuzende Schnitte minimalen Gewichts λ in G
- $\Rightarrow$ $c(A, D) = c(B, C) = 0$ und
  $c(A, B) = c(B, D) = c(D, C) = c(C, A) = \lambda/2$
- $\gg$ Sind s und t zwei adjazente Knoten mit $c(\{s, t\}) > 0$, so enthält die Menge der minimalen Schnitte von G, die s und t trennen, keine zwei sich kreuzenden Schnitte.

Algorithmus von Stoer und Wagner

Vorgehen

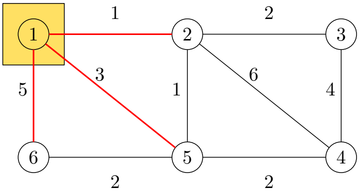
- $|V| - 1$ Phasen
- in Phase i wird Schnitt der Phase i berechnet
- Schnitt der Phase ist minimaler s-t-Schnitt

Schnitt der Phase i

- beginne mit beliebigem Startknoten a und setze $S_i := \{a\}$
- wähle nächsten Knoten v_j so, dass $c(S_i, v_j)$ maximiert wird
- setze $S_i := S_i \cup \{v_j\}$ $(j < n)$
- seien $s := v_{n-1}, t := v_n$ die letzten beiden Knoten
- Schnitt der Phase i ist $(S_i, V \setminus S_i) = (\{t\}, V \setminus \{t\})$
- kontrahiere s und t

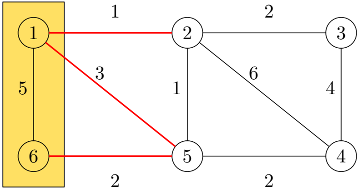
Aufgabe 2

Phase 1



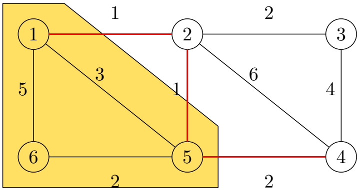
Aufgabe 2

Phase 1



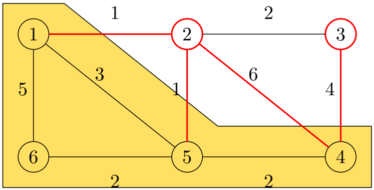
Aufgabe 2

Phase 1



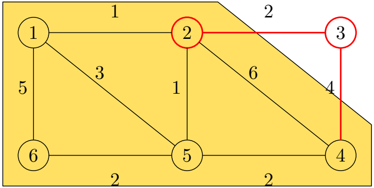
Aufgabe 2

Phase 1



Aufgabe 2

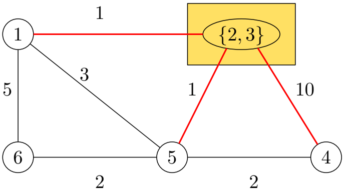
Phase 1



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Aufgabe 2

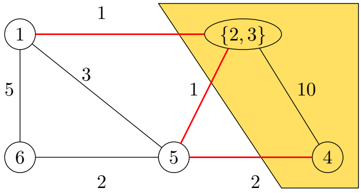
Phase 2



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Aufgabe 2

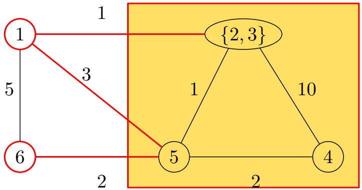
Phase 2



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Aufgabe 2

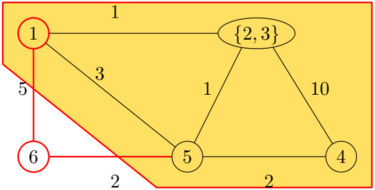
Phase 2



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Aufgabe 2

Phase 2

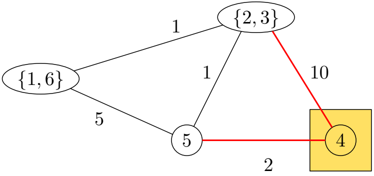


Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Aufgabe 2

Phase 3

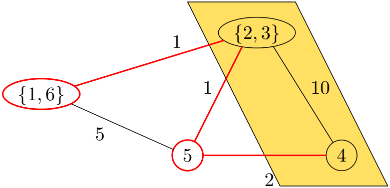


Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Aufgabe 2

Phase 3

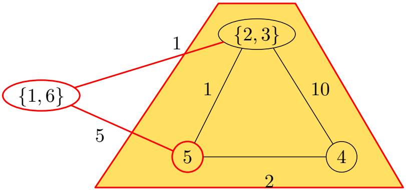


Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Aufgabe 2

Phase 3



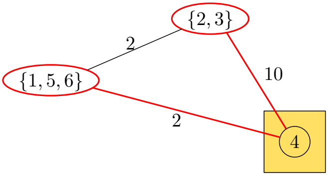
Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Schnitt der Phase 3: $S_3 = \{1, 6\}$, $w = 6$

Aufgabe 2

Phase 4



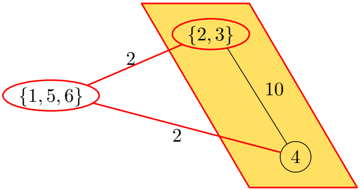
Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Schnitt der Phase 3: $S_3 = \{1, 6\}$, $w = 6$

Aufgabe 2

Phase 4



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

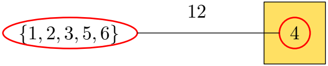
Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Schnitt der Phase 3: $S_3 = \{1, 6\}$, $w = 6$

Schnitt der Phase 4: $S_4 = \{1, 5, 6\}$, $w = 4$

Aufgabe 2

Phase 5



Schnitt der Phase 1: $S_1 = \{3\}$, $w = 6$

Schnitt der Phase 2: $S_2 = \{6\}$, $w = 7$

Schnitt der Phase 3: $S_3 = \{1, 6\}$, $w = 6$

Schnitt der Phase 4: $S_4 = \{1, 5, 6\}$, $w = 4$

Schnitt der Phase 5: $S_5 = \{4\}$, $w = 12$

Aufgabe 2

(b) An welcher Stelle im Korrektheitsbeweis zum Algorithmus von Stoer und Wagner wurde verwendet, dass die Kantengewichte nicht negativ sind?

Finden Sie ein Beispiel eines Graphen mit negativen Kantengewichten und einen Startknoten a, so dass der Algorithmus von Stoer und Wagner keinen minimalen Schnitt liefert.

Aufgabe 2

- $Z := (S'_u, (V \setminus S') \cap (S_u \cup \{u\}))$
- $A := (S'_v, (V \setminus S') \cap (S_v \cup \{v\}))$
- $B := (S_u \setminus S_v, u)$

In der Vorlesung wurde gezeigt

- $A \subseteq Z$
- $B \subseteq Z$
- $A \cap B = \emptyset$

Abschätzung im Induktionsschritt

- $c(S_u, u) \leq c(A) + c(B) \leq c(Z)$
- gilt nur, falls $c(Z \setminus (A \cup B)) \geq 0$

Aufgabe 3

Aufgabe (Flüsse)

Betrachte Flussnetzwerk $D = (V, A)$ mit Rückwärtskanten, d.h. es gibt Kanten $(v, u) \in A$ mit $(u, v) \in A$.

- Zeigen Sie, dass Flussnetzwerk $D' = (V, A')$ ohne Rückwärtskanten existiert, so dass $A' \subseteq A$.

Klassisches Flussnetzwerk

Sei $(D; s, t; c)$ Netzwerk. Abbildung $f : E \longrightarrow \mathbb{R}_0^+$ heißt Fluss, wenn sie die folgenden Eigenschaften erfüllt:

(1) Für alle $(i, j) \in E$ ist die Kapazitätsbedingung

$$0 \leq f(i,j) \leq c(i,j) \tag{5}$$

erfüllt.

(2) Für alle $i \in V \setminus \{s, t\}$ ist die Flusserhaltungsbedingung

$$w(i) := \sum_{\{j|(i,j)\in E\}} f(i,j) - \sum_{\{j|(j,i)\in E\}} f(j,i) = 0 \tag{6}$$

Wert des Flusses

$$w(f) := w(s) = \sum_{(s,i)\in E} f(s,i) - \sum_{(i,s)\in E} f(i,s)$$

Flussnetzwerk von Goldberg-Tarjan

Motivation

- $\gg$ betrachten modifizierte Definition eines Flussnetzwerkes
- $\rightsquigarrow$ Analyse verständlicher
- $\gg$ Idee: Schiebe Präfluss in Richtung Senke und schiebe Überschuss auf Rückwärtskanten E_R wieder zurück zur Quelle
- $\rightsquigarrow$ betrachten Graphen $D' = (V, E')$ mit

$$E' = E \cup E_R$$

  mit $E_R = \{(v, w) \mid (w, v) \in E, (v, w) \notin E\}$
- $\gg$ $c(e) := 0$ für $e \in E_R$

Flussnetzwerk von Goldberg-Tarjan

Kapazitätsbedingung

- Fluss kann negativ sein (Rückfluss auf Gegenkante)

$$\forall (v, w) \in V \times V \quad f(v, w) \leq c(v, w)$$

- Bemerkung: für $e \in E_R$ gilt $f(e) \leq c(e) = 0$

Antisymmetrie-Forderung

- Rückfluss auf Gegenkante entspricht Fluss auf Vorwärtskante

$$\forall (v, w) \in V \times V \quad f(v, w) = -f(w, v)$$

Flussnetzwerk von Goldberg-Tarjan

Flusserhaltungsbedingung

- modifizierte Flusserhaltungsbedingung

$$\forall v \in V \setminus \{s, t\} \quad \sum_{u \in V} f(u, v) = 0$$

Der Wert eines Flusses f ist dann

$$w(f) = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t) \ .$$

Flussnetzwerk von Goldberg-Tarjan

Aufgabe (Wohldefiniertheit)

Zeigen Sie: Ist f Fluss nach Goldberg-Tarjan auf D', so ist $f_{|D}$ ein klassischer Fluss auf D

Algorithmus von Goldberg-Tarjan

Datenstrukturen

- Distanz- bzw. Höhenlabel $\text{dist} : V \to \mathbb{N}$
- Präfluss $f : V \times V \to \mathbb{R}$
- Flussüberschuss $e : V \to \mathbb{R}$
- Residualkapazität $r_f : V \times V \to \mathbb{R}$

Initialisierung

- $\text{dist}(s) = |V|$, $\text{dist}(v) = 0 \quad (v \neq s)$
- $r_f(v, w) = c(v, w)$
- $f(s, v) = c(s, v), \quad r_f(s, v) = 0$
- $f(v, s) = -c(s, v), \quad r_f(s, v) = c(v, s) - f(v, s)$
- $e(v) = c(s, v)$ für v inzident zu v

Algorithmus von Goldberg-Tarjan

Vorgehen

- Knoten $v \neq s, t$ heißt aktiv, falls $e(v) > 0$ und $\text{dist}(v) < \infty$
- solange aktive Knoten existieren: wähle aktiven Knoten und wende Push oder Relabel an

Bemerkung

- Wahl des aktiven Knotens kann sich stark auf die Laufzeit auswirken
- Strategien: FIFO, Highest-Label, Excess-Scaling

Algorithmus von Goldberg-Tarjan

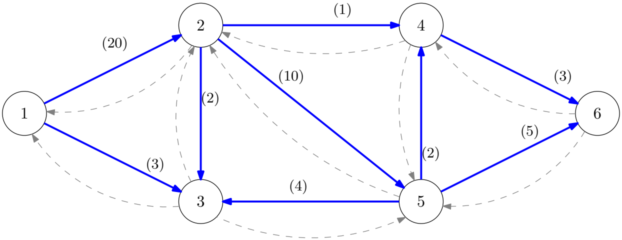
Push

- $\textsc{Push}(u, v)$ anwendbar, falls u aktiv, d.h. $e(u) > 0$, $r_f(u, v) > 0$ und $\text{dist}(u) = \text{dist}(v) + 1$
- Effekt: Knotenüberschuss wird nach v geschoben
- Änderungen: $f(u, v)$, $f(v, u)$, $r_f(u, v)$, $r_f(v, u)$, $e(u)$, $e(v)$

Relabel

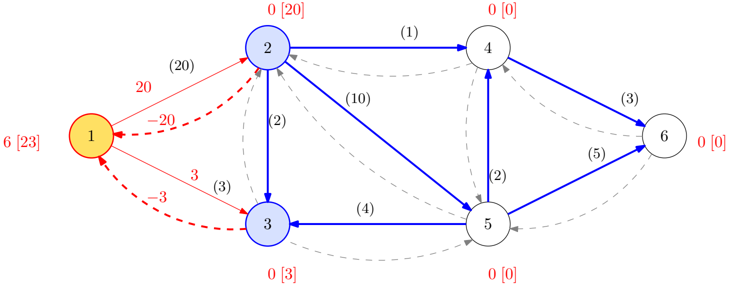
- $\textsc{Relabel}(u)$ anwendbar, falls u aktiv und für alle v mit $r_f(u, v) > 0$ gilt $\text{dist}(u) \leq \text{dist}(v)$
- d.h. kein Push anwendbar
- Effekt: Distanzlabel von u wird inkrementiert
- Änderungen: $\text{dist}(u)$

Aufgabe 4



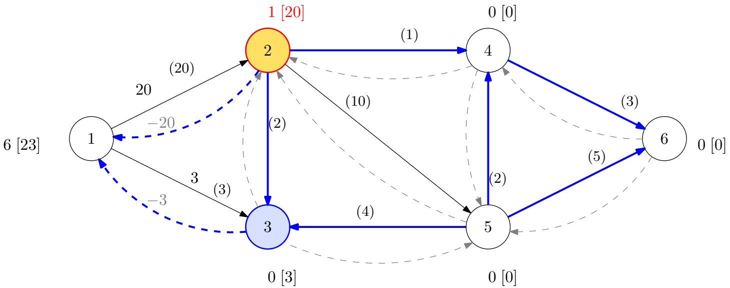
Aufgabe 4

Initialisierung



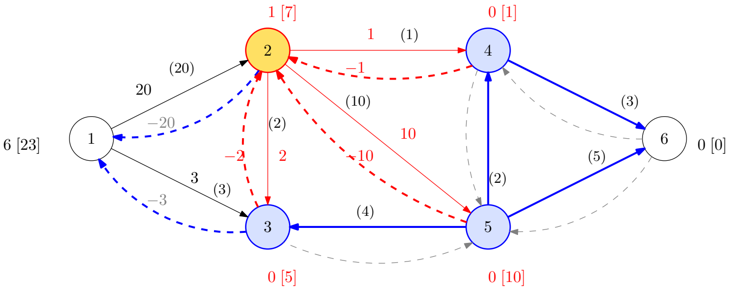
Aufgabe 4

1. RELABEL(2)



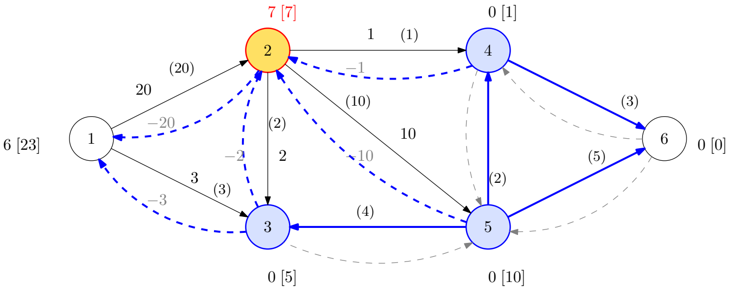
Aufgabe 4

2. $\textsc{Push}(2,3)$, $\Delta = 2$, $\textsc{Push}(2,4)$, $\Delta = 1$,$\textsc{Push}(2,5)$, $\Delta = 10$



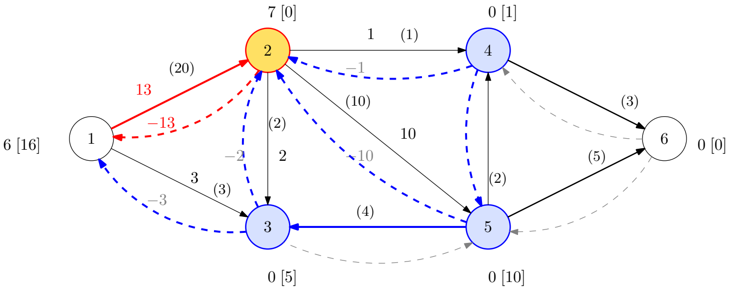
Aufgabe 4

3. Relabel(2)



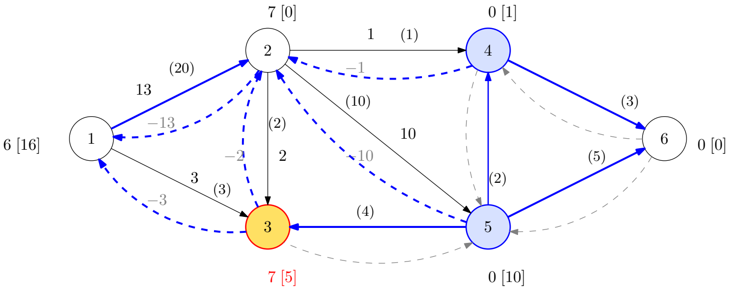
Aufgabe 4

4. $\text{Push}(2,1)$, $\Delta = 7$



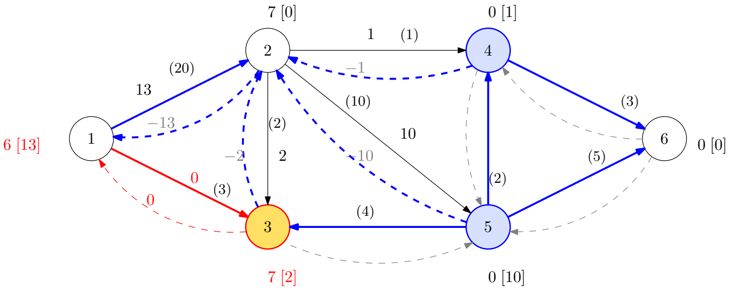
Aufgabe 4

5. Relabel(3)



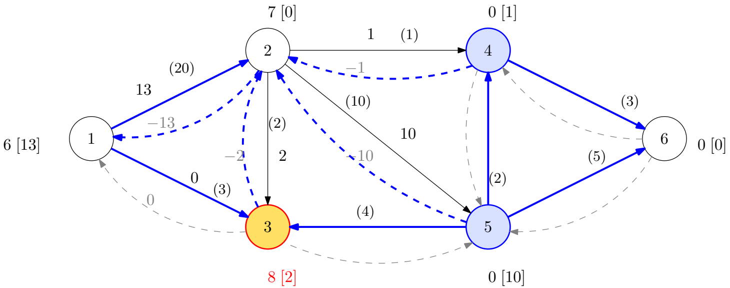
Aufgabe 4

6. $\text{Push}(3,1)$, $\Delta = 3$



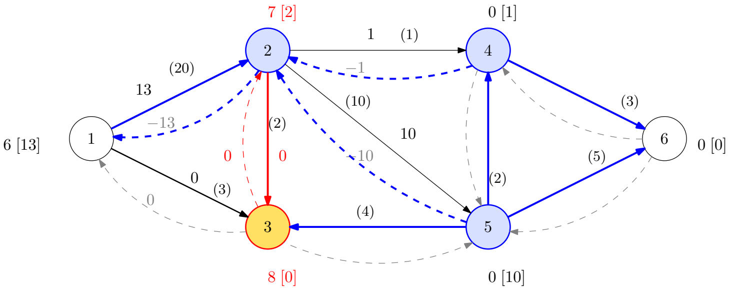
Aufgabe 4

7. RELABEL(3)



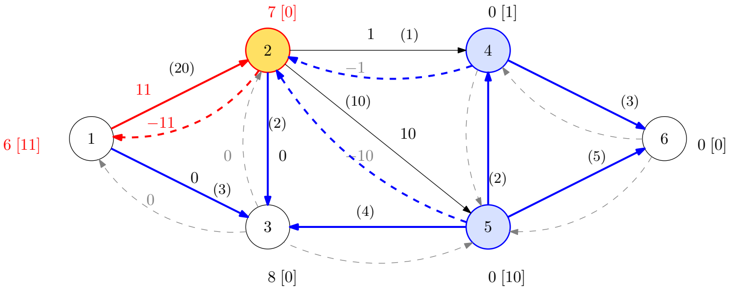
Aufgabe 4

8. Push$(3, 2)$, $\Delta = 2$



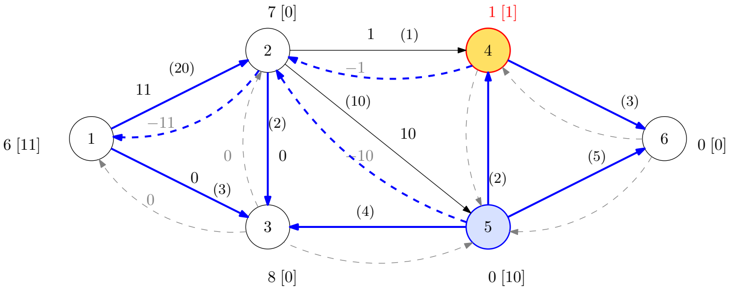
Aufgabe 4

9. PUSH$(2, 1)$, $\Delta = 2$



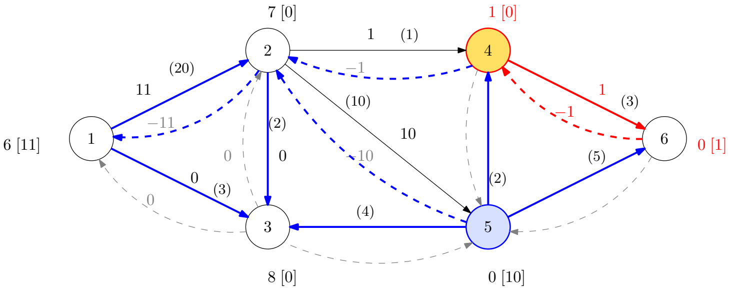
Aufgabe 4

10. Relabel(4)



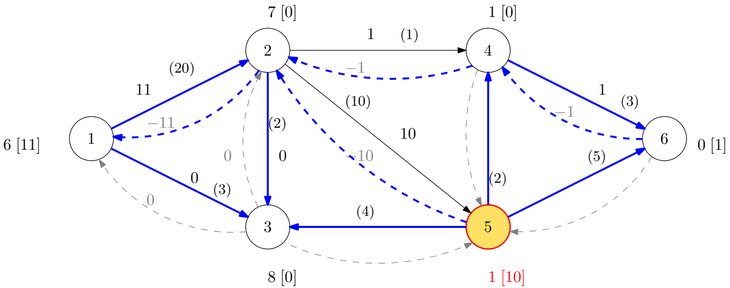
Aufgabe 4

11. Push$(4, 6)$, $\Delta = 1$



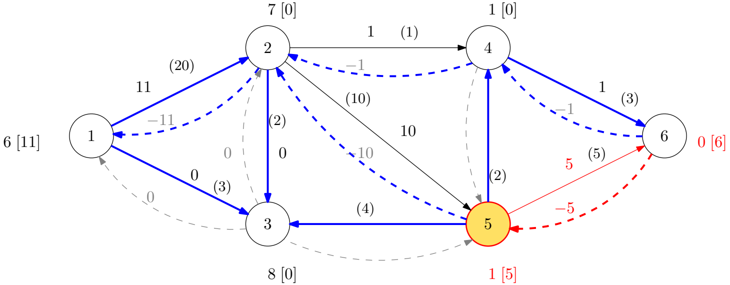
Aufgabe 4

12. Relabel(5)



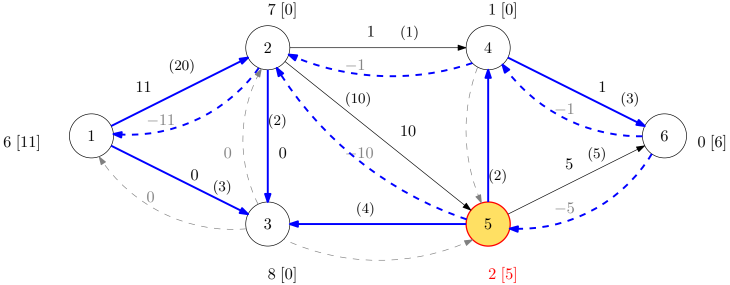
Aufgabe 4

13. Push$(5, 6)$, $\Delta = 5$



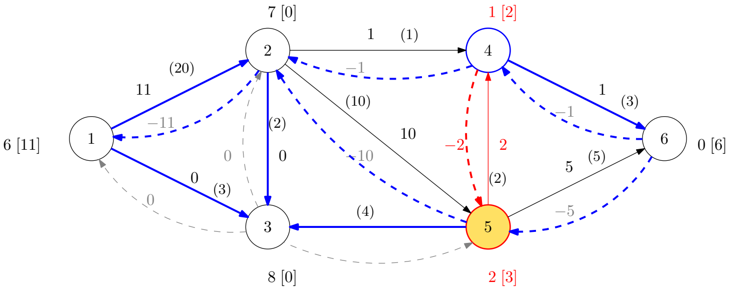
Aufgabe 4

14. Relabel(5)



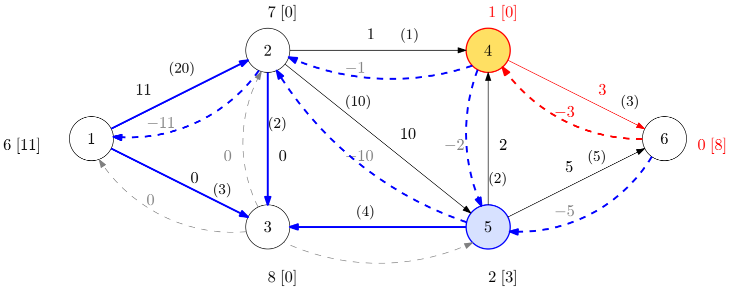
Aufgabe 4

15. PUSH$(5, 4)$, $\Delta = 2$



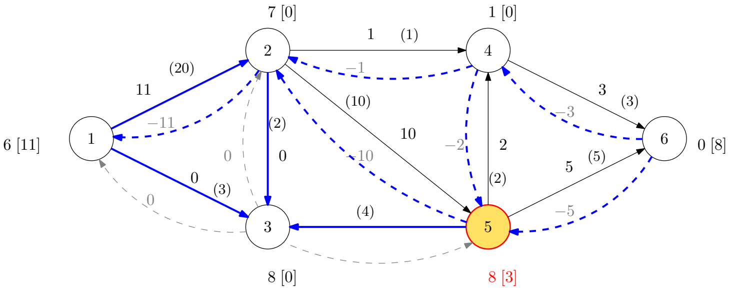
Aufgabe 4

16. PUSH(4, 6), $\Delta = 2$



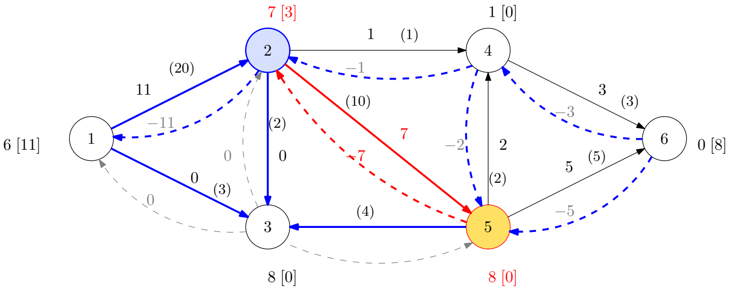
Aufgabe 4

17. RELABEL(5)



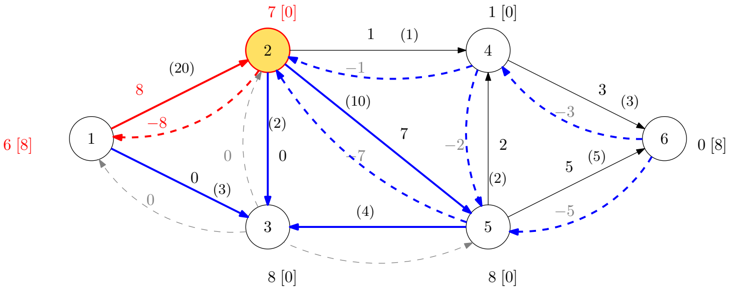
Aufgabe 4

18. $\text{Push}(5, 2)$, $\Delta = 3$



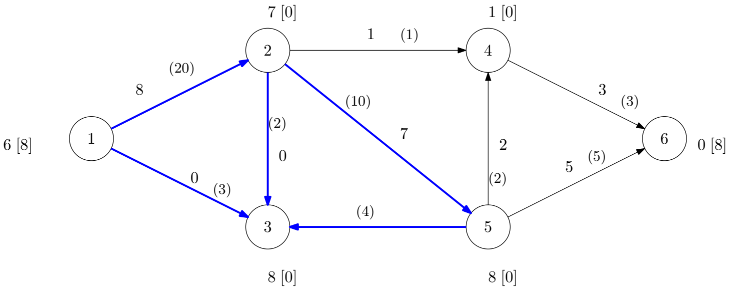
Aufgabe 4

18. $\text{PUSH}(2,1)$, $\Delta = 3$



Aufgabe 4

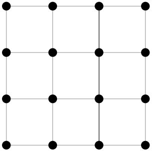
Ausgabe



Aufgabe 5

Escape Problem

Gegeben: $n \times n$ Gitterpunkte Γ, $S \subseteq \Gamma$ mit $|S| < n^2$.



Frage: Gibt es $|S|$ viele knotendisjunkte Wege gibt, so dass jeder Weg von einem Knoten aus S startet und an einem Randpunkt des Gitters endet.

Aufgabe 5

Aufgabe (Das Escape Problem)

(a) Geben Sie für $n = 4$ eine Ja-Instanz mit möglichst vielen Startknoten an, sowie eine Nein-Instanz mit möglichst wenig Startknoten.

Aufgabe 5

Aufgabe (Das Escape Problem)

Betrachte Flussnetzwerk mit Knotenkapazitäten $\gamma : V \to \mathbb{R}_0^+$ so dass

- Für alle $i \in V$ ist die *Knotenkapazitätsbedingung*

$$\sum_{\{j | (i,j) \in E\}} f(i,j) \leq \gamma(i) \quad \text{wenn} \quad i \in V \setminus \{t\}$$
$$\sum_{\{j | (j,i) \in E\}} f(j,i) \leq \gamma(i) \quad \text{wenn} \quad i = t$$

  erfüllt.

(b) Führen Sie das Flussnetzwerk auf ein äquivalentes ohne Knotenkapazitäten zurück.

Aufgabe 5

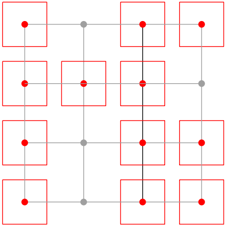
Aufgabe (Das Escape Problem)

(c) Wie würden Sie das Escape-Problem algorithmisch lösen? Geben Sie die Worst-case Laufzeit Ihres Ansatzes an.

Aufgabe 5

Schritt 1

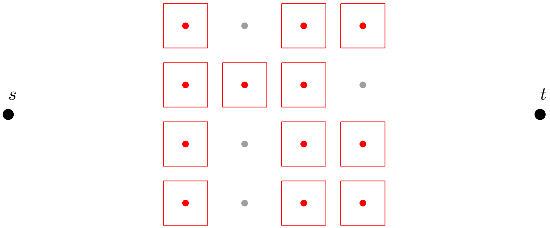
- konstruieren Flussnetzwerk $\mathcal{N}$ mit Knotenkapazität $\gamma \equiv 1$ aus dem Gitter wie folgt:



Aufgabe 5

Schritt 1

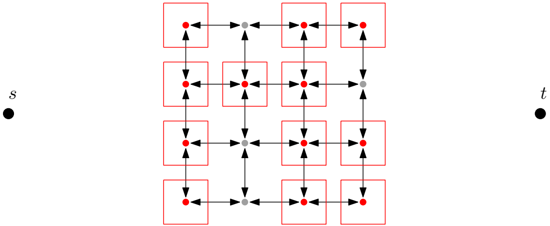
≫ Knotenmenge ist Menge der Gitterpunkte und zuätzlich eine Quelle s und Senke t



Aufgabe 5

Schritt 1

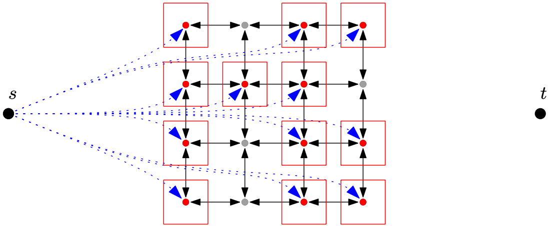
≫ inzidente Gitterpunkte werden bidirektional verbunden (Kantenkapazität ist 1)



Aufgabe 5

Schritt 1

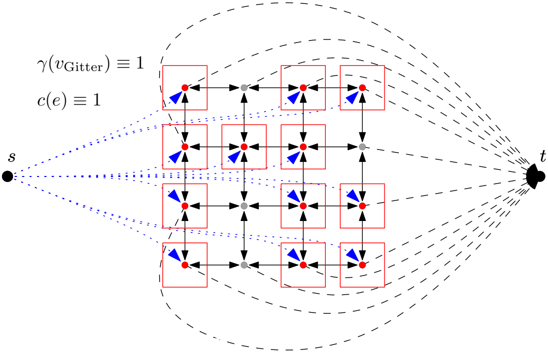
≫ füge Kanten (s, v) mit Kapazität 1 für $v \in S$ (Menge der Startknoten) hinzu



Aufgabe 5

Schritt 1

≫ füge Kanten (r, t) mit Kapazität 1 für alle Randknoten r hinzu



≫ Fluss in $\mathcal{N}$ induziert knotendisjunkte Wege

Aufgabe 5

Größe des Flussnetzwerkes

- $|V| = n^2 + 2 \in \mathcal{O}(n^2)$ Knoten
- $|E| = |S| + 4(n-1) + 2 \cdot 2n(n-1) \in \mathcal{O}(n^2)$ Kanten

genauer

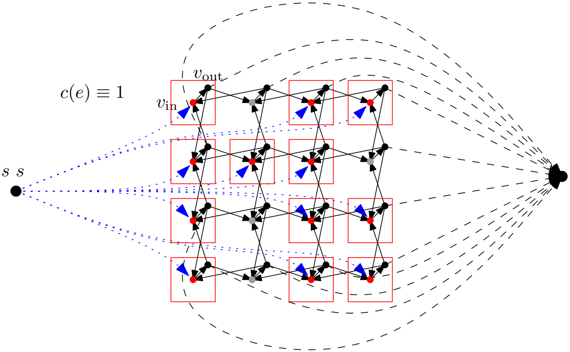
- $|S| < n^2$ Kanten von der Quelle zu den Startknoten
- $4(n-1)$ Kanten von Randknoten zur Senke
- $2 \cdot 2(n-1)$ je 2 Kanten zwischen den Gitterpunkten

$\Rightarrow$ Flussnetzwerk kann in Zeit $\mathcal{O}(n^2)$ konstruiert werden.

Aufgabe 5

Schritt 2

≫ Wandele $\mathcal{N}$ in äquivalentes Flussnetzwerk $\mathcal{N}'$ ohne Knotenkapazitäten um (Aufgabenteil b))



Aufgabe 5

Größe des modifizierten Flussnetzwerks

- $|E'| = |E| + |V| \in \mathcal{O}(n^2)$ Kanten
- $|V'| = 2|V| \in \mathcal{O}(n^2)$ Knoten

Aufgabe 5

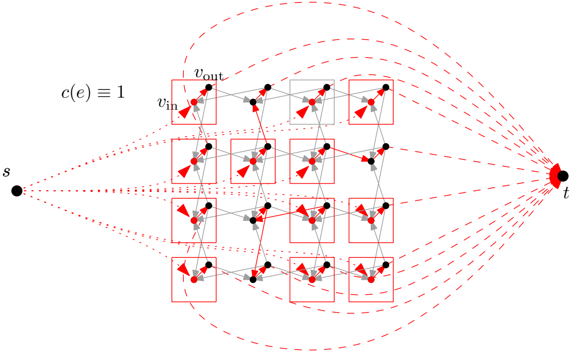
Schritt 3

- Berechne maximalen Fluss f.
- Es gibt genau dann $|S|$ knotendisjunkte Wege, wenn $w(f) = |S|$ gilt.
- Größe des Flussnetzwerkes: $|E'|, |V'| \in \mathcal{O}(n^2)$.

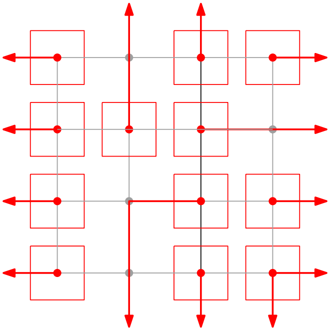
Laufzeit

- Goldberg-Tarjan: $\mathcal{O}(|V'|^2|E'|) = \mathcal{O}(n^6)$
- Goldberg-Tarjan (Highest-Label): $\mathcal{O}(|V'|^2|E'|^{1/2}) = \mathcal{O}(n^5)$
- Goldberg-Tarjan (Excess-Scaling): $\mathcal{O}(|E'| + |V'|^2 \log C) = \mathcal{O}(n^4)$ (C konstant)
- Ford-Fulkerson: $\mathcal{O}(|E'| \cdot |f^*|) = \mathcal{O}(|E'| \cdot 4(n-1)) = \mathcal{O}(n^3)$

Aufgabe 5



Aufgabe 5



Aufgabe 5

Bemerkung

- Flussnetzwerk kann verwendet werden, um maximale Menge von Startknoten zu finden:
- wähle dazu $S = \Gamma$
- konstruiere Flussnetzwerk zu S
- berechne maximalen Fluss