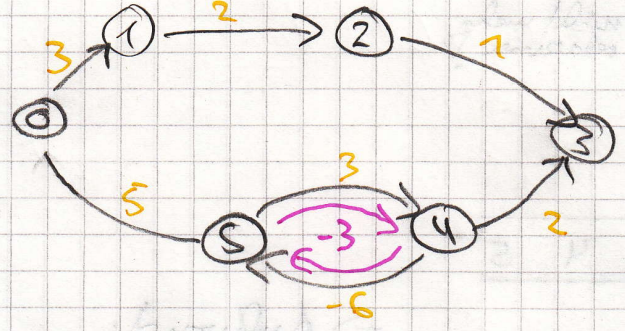
Kürzeste Pfade

Bsp.: Zyklen mit negativer Länge



Zyklen mit negativen Gewicht?

Kürzester Pfad von Knoten ⓪ nach ③? → existiert nicht, wegen negativen Zyklen.

Omega → $\omega(0,1,2,3) = 3+2+1 = 6$

$\omega(0,5,4,3) = 5+3+2 = 10$

Relaxieren einer Kante

Sit1

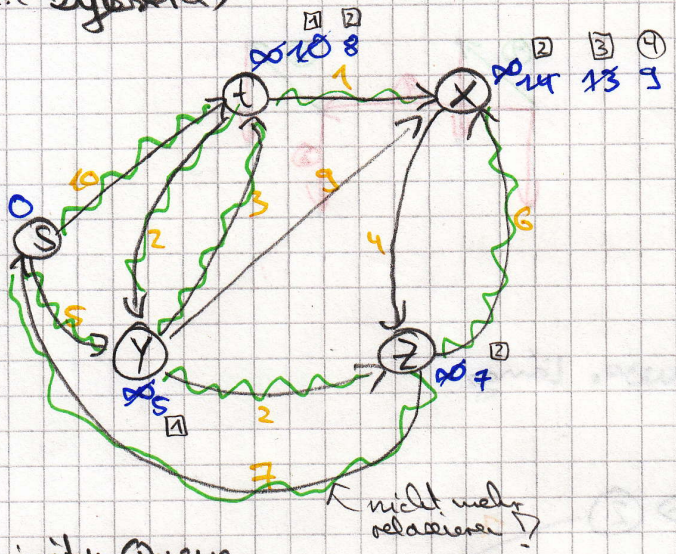
Attribute d: bisher gefundener minimaler Abstand

relax(u,v) → erfordert Update von v.d auf 7

Sit 2

relax(u,v) erfordert kein update

Bsp.: (Dijkstra)



Priority Queue

Knoten	Schritt 0	1	2	3	4	5
s	0	X				
t	∞	10	8		X	
x	∞		14	13	9	X
y	∞	5	X			
z	∞		7	X		

⇒ Aufwand:

$$|V| \cdot \begin{pmatrix} q.\text{insert} \\ q.\text{dataMin} \\ q.\text{findMin} \end{pmatrix} + O(|E|)$$

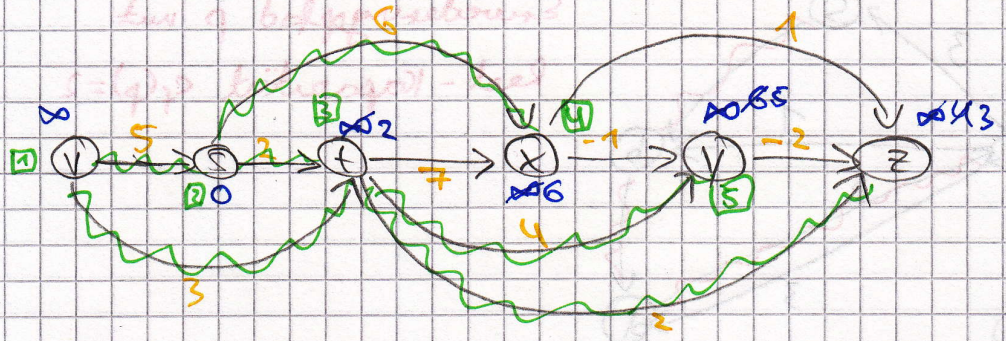
$$\Uparrow$$

$$O(\log(|V|))$$

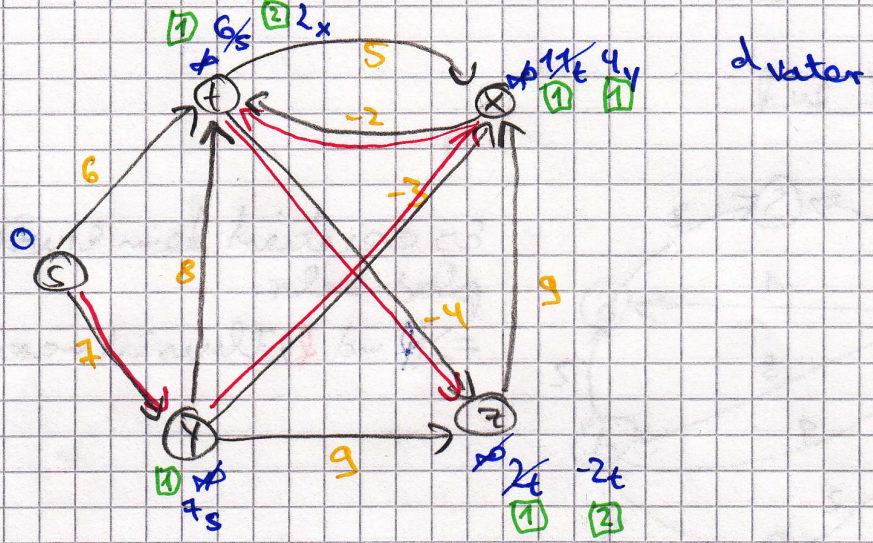
Gesamt:

$$O(|V| \cdot \log(|V|) + |E|)$$

Bsp. (DAG-Algorithmus)



Bsp (Bellman-Ford)



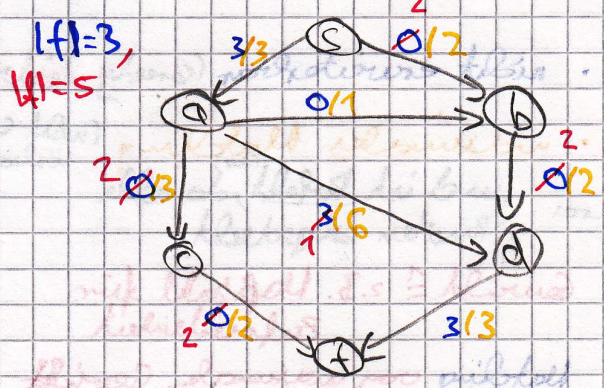
Reihenfolge, in der Kanten relaxiert werden:

- durch Knoten gehen s, t, x, y, z
- Kanten aufsteigend nach Zielknoten

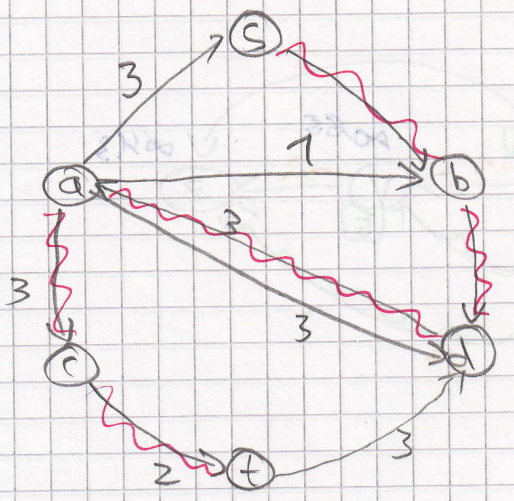
Kürzester Weg von (s) nach (z) (Rückwärts auflösen)

(s, y, x, t, z)

Bsp (Maximaler Fluss)



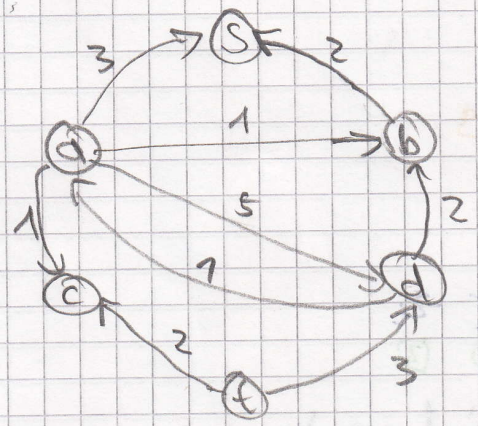
Restnetzwerk G_f



mit Tiefen- oder Breitensuche

Erweiterungspfad p mit Rest-Kapazität $c_f(p) = 2$

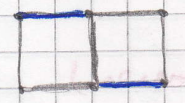
Restnetzwerk zu f



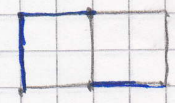
Es existiert kein Erweiterungspfad mehr.
$\Rightarrow$ (f und f) Fluss ist maximal

Matching

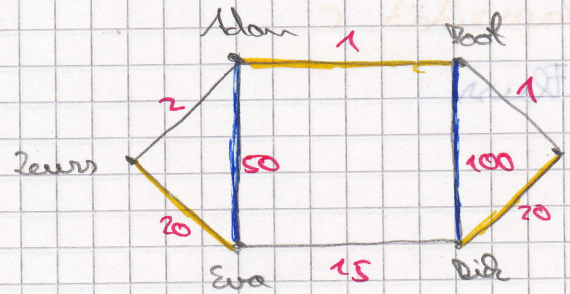
M ist Matching



Kein Matching



Anwendung: Reiseteilnehmer

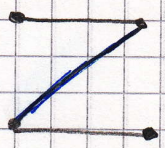


- nicht erweiterbar (greedy lösbar)
- maximales Matching (nicht greedy lösbar) (*)
  und ist perfekt, da alle Knoten abgedeckt.

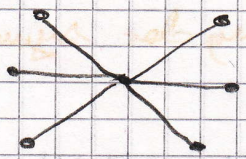
Gewicht $\hat{=}$ z.B. Maßzahl für Zufriedenheit

Matching von maximalem Gewicht

(*)

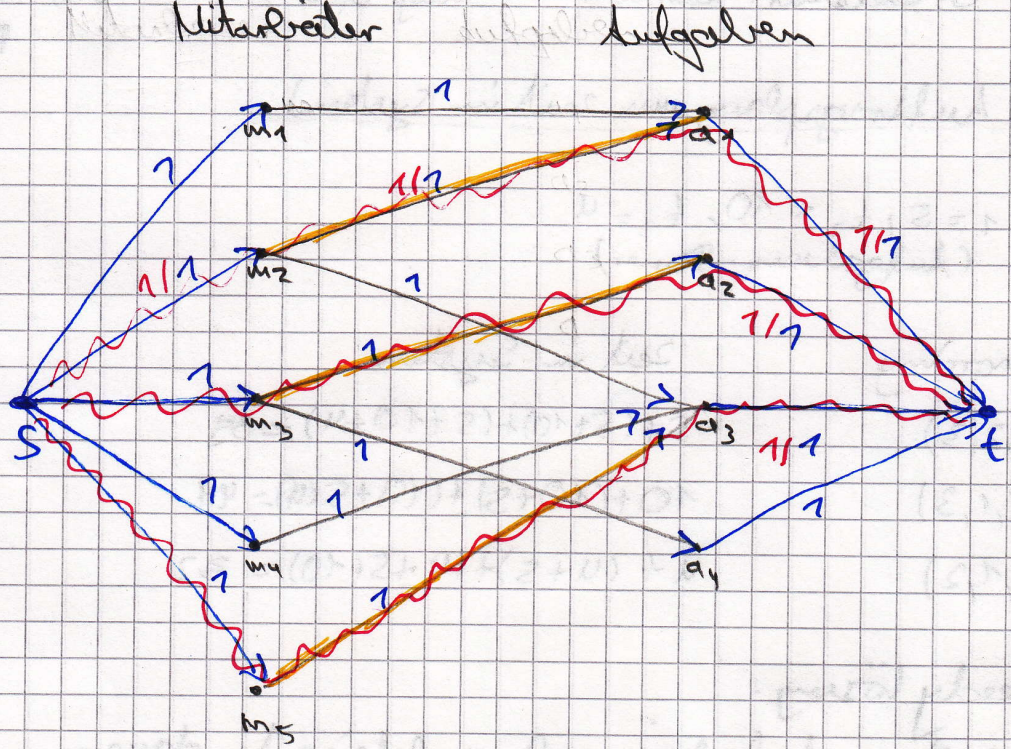


(**)



besitzt kein perfektes Matching

Bipartites Matching



maximaler Fluss ($\Rightarrow$ maximales Matching

V_1 V_2

Überführung in maximals Fluss-Problem

nach Heirats-Satz kann kein Matching M mit $|M| = |V_1|$ existieren, denn

$A = \{m_4, m_5\}$, $|A| = 2$ $\vee$

$N(A) = \{a_3\}$, $|N(A)| = 1$

Allgemeiner Lösungs-Ansatz:

Start-Matching
~ augmentierender Pfad
durch Bildung der symmetrischen Differenz
$M' = M \triangle P$
$= (M \setminus P) \cup (P \setminus M)$
(Vergrößerung des Matchings um 1 Kante)

Es existiert kein weiterer aug. Pfad ⇒ maximales (hier sogar Perfekt)

Bsp (Auftragsplanung mit Zeit im System)

$t_1 = 5, t_2 = 10, t_3 = 4$
(Aufgaben-Dauer)

Anordnung	Zeit im System
(1,2,3)	$5+(5+10)+(5+10+4) = 39$
(2,1,3)	$10+(10+5)+(10+5+4) = 44$
(3,1,2)	$4+(4+5)+(4+5+10) = 32$

Greedy Lösung:
① Sortiere Aufgaben nach aufsteigender Dauer
② Abarbeitung in dieser Reihenfolge

Verarbeitung auf m Servern:

① Sortieren $t_1, t_2, \ldots t_n$
← wird zuerst fertig
Ser 1 t_1 t_{m+1} ← wird als erste 2te Aufgabe bearbeitet und fertig
Ser 2 t_2 ⋮
Ser 3 t_3 ⋮
Server m t_m t_{2m}

Insgesamt werden die Aufgaben der optimalen Reihenfolge $t_1, t_2, \ldots t_n$ fertig

Bsp (Unit-Time-Tasks)

Aufg.	Schlusstermin	Gewinn
1	2	30
2	1	35
3	2	25
4	1	40

Anordnung (1,2) wäre nicht zulässig

(3,1) zulässig (Gewinn: 25+30=55)

Optimal?

↳ Greedy: ① Sortieren nach fallenden Gewinn

② Hinzufügen, wenn Auswahl zulässig bleibt

$I=\{4\}$, $I=\{4,2\}$ nicht zulässig

$I=\{4,1\}$ zulässig

$I=\{4,3\}$ nicht zulässig

⇒ Optimal (4,1) → Gewinn=70