(iv) Wir brauchen zunächst zwei Eigenschaften der Warteschlange Q:

(a) $\ell(v)$ wächst monoton mit der Aufnahme von v in die Warteschlange Q.

und (noch stärker)

(b) Zu jedem Zeitpunkt gilt für die Warteschlange

$Q: v_i, v_{i+1}, \dots, v_k$, dass

$\ell(v_i) \le \dots \le \ell(v_k) \le \ell(v_i)+1$. (Also: Die Warteschlange hat höchstens zwei verschiedene ℓ-Werte.)

Beweis durch Induktion über die Zahl z der aufgenommenen Kanten.

Induktionsanfang: $z=0$

Hier besteht die Warteschlange nur aus s, beide Eigenschaften gelten.

Induktionsannahme:

Die Aussage gelte nach z Kanten. Sei

$Q: v_i, v_{i+1}, \dots, v_k$ und

$$\ell(v_i) \le \ell(v_{i+1}) \le \dots \le \ell(v_k) \le \ell(v_i)+1$$

Induktionsschritt:

Wir fügen eine Kante ein; werden dafür vorne Knoten aus Q gelöscht (weil sie keine neuen Nachbarn haben), ändert das nichts an der Monotonie (a) und Eigenschaft (b). Sei danach v_j der erste Knoten, ~~und sei v_j~~ also der Knoten, für den die neue Kante $e=\{v_j, v_{k+1}\}$ eingefügt wird, ~~dann~~ bekommt man:

$$Q: \quad v_j, v_{j+1}, \ldots, v_k, v_{k+1}$$

mit

$$\ell(v_i) \le \ldots \le \ell(v_j) \le \ldots \ell(v_k) \le \ell(v_i)+1 \;.$$

Wegen $\ell(v_i) \le \ell(v_j)$ gilt auch $\ell(v_i)+1 \le \ell(v_j)+1 = \ell(v_{k+1})$.

Damit erhält man

$$\ell(v_j) \le \ldots \le \ell(v_k) \le \ell(v_i)+1 \le \ell(v_j)+1 = \ell(v_{k+1})$$

und sowohl (a) als auch (b) gelten weiterhin.

Jetzt nehmen wir an, am Ende des Algorithmus gibt es einen Knoten $w \in V$ mit

$$d(s,w) < d_{(R,T)}(s,w) = \ell(w) \,.$$

Unter den Knoten mit dieser Eigenschaft wählen wir einen mit minimalem Abstand von s in G.

Sei P ein kürzester s-w-Weg in G, und sei $e = \{v, w\}$ die letzte Kante in P, d.h.

$$d(s,w) = d(s,v) + 1$$

Wir haben $d(s,v) = d_{(R,T)}(s,v)$,

aber $d(s,w) < d_{(R,T)}(s,w)$,

also gehört e nicht zu T.

Außerdem ist

$$\ell(w) = d_{(R,T)}(s,w) > d(s,w) = d(s,v)+1$$
$$= d_{(R,T)}(s,v)+1 = \ell(v)+1.$$

Wäre $w \in R$, hätten wir wegen

$$\ell(w) > \ell(v)+1$$

einen Widerspruch zur Eigenschaft $(*)$

Also muss $w \notin R$ gelten; dann verbindet aber die Kante $e = \{v,w\}$ zum Zeitpunkt der Entfernung von v aus Q v mit einem Knoten $w \notin R$, im Widerspruch zur Abfrage in 3.

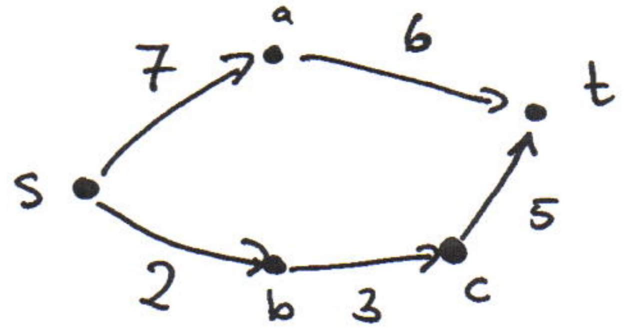
$\square$

3.8 Ausblick: Algorithmische Probleme auf Graphen

Problem 1: Kürzeste Wege mit Kantenlängen

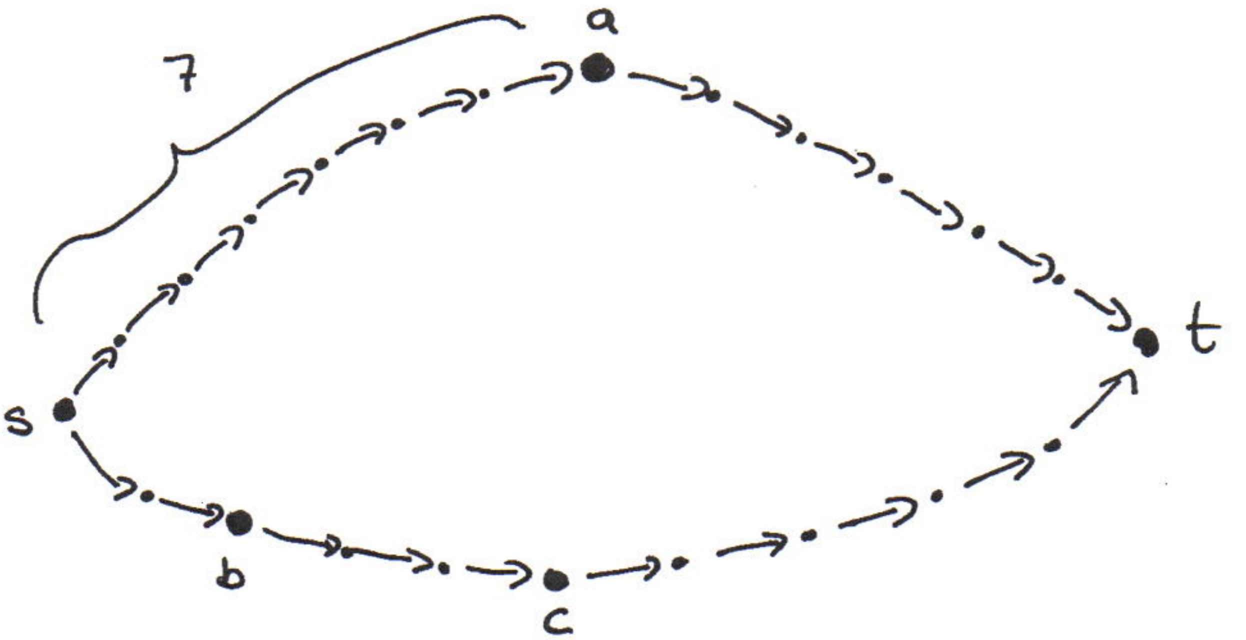
Gegeben: Graph $G=(V,E)$, $c: E \to \mathbb{N}$ Kantenlänge,
zwei Knoten $s,t \in V$

Gesucht: Kürzester Weg von s nach t.



Würde man die Kantenlängen ignorieren, also einen Weg von s nach t mit den wenigsten Kanten suchen, und BFS anwenden, wäre das Ergebnis 13 (und das ist nicht optimal!)

Ersetzt man eine Kante der Länge $c(e)$ durch $c(e)$ Kanten der Länge 1, könnte man auf diesen Graphen BFS anwenden.



Allerdings wird in diesem Graphen die Laufzeit von BFS ziemlich schlecht. [Abhängig von ~~Gesamtlängen~~ Kantenlängen!]

Problem 2: Kostengünstigste Netzwerke

Gegeben: Graph $G=(V,E)$. $c: E \to \mathbb{N}$ Kantengewicht.

Gesucht: Ein zusammenhängender Teilgraph T mit
$\sum_{e \in T} c(e)$ minimal.

Hat man positive Kantengewichte, gibt es in der Lösung nie einen Kreis (man könnte eine Kante weglassen und hätte eine bessere Lösung)

Deshalb spricht man auch von:
Minimalen aufspannenden Bäumen (MSTs).

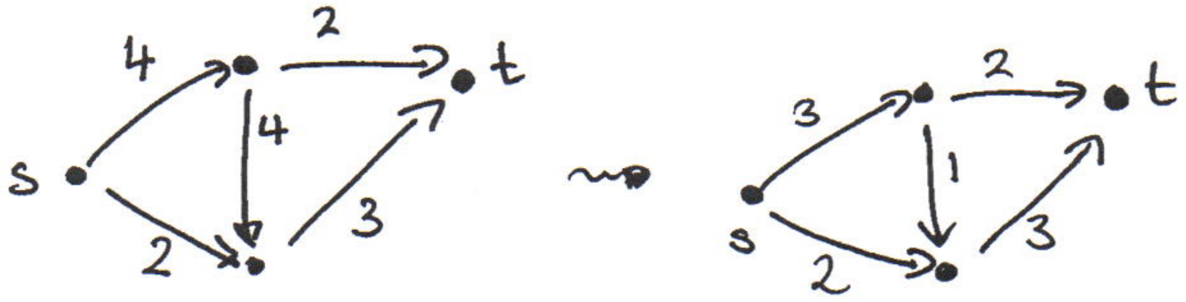
Problem 3: Flussprobleme

Gegeben: Graph $G=(V,E)$. Kapazitäten der Kanten: $u: E \to \mathbb{N}$. 61
Zwei Knoten s,t.

Gesucht: Maximaler Fluss von s nach t
(der die Kapazitäten auf den Kanten nicht überschreitet)

Kapazität

Fluss



Knoten $v \in V \setminus \{s,t\}$ leiten das Gut nur weiter.

Beispiele
- Verkehr (evtl. mehrere „Quellen")
- Datenpakete
- Lieferkette