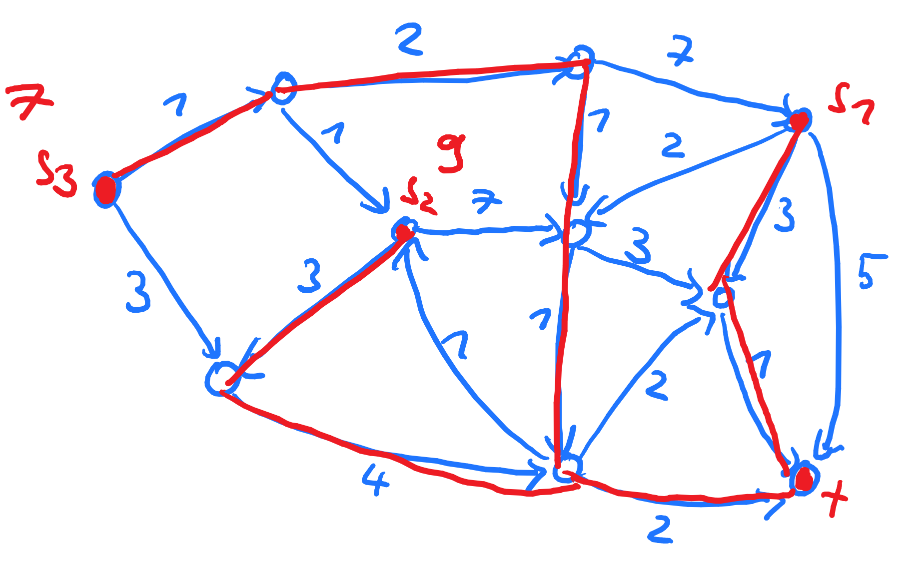


Algo:

Drehe Richtung aller Kanten um.

Also: aus (u,v) wird (v,u)

Auf dem Resultierenden Graphen:

Dijkstra von t. ⇝ Distanzen bei den $s \in S$ sind die gewünschten Distanzen.

Korrektheit: Gerichtete st-Pfade drehen sich um zu gerichteten ts-Pfaden.

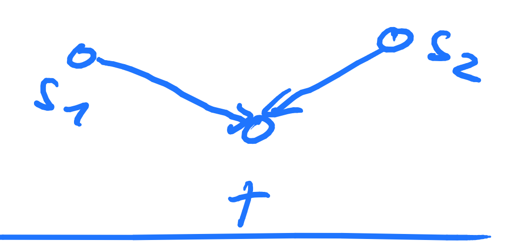
⇒ Auf dem resultierenden Graphen müssen wir die kürzesten ts-Pfade finden für alle $s \in S$. ⇝ das tut Dijkstra.

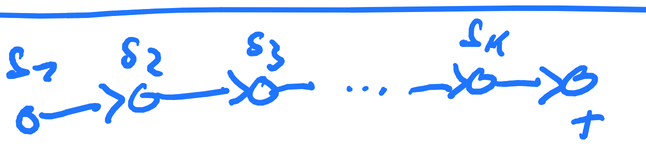
Laufzeit: $O(m+n)$ fürs Kanten Umdrehen

$+ \Theta((n+m) \cdot \log n)$ für Dijkstra

(mit binärem Heap)

$\Rightarrow \Theta((n+m)\log n)$





In dieser Familie von Graphen ändert sich die Distanz für jedes s_i, wenn wir die Länge von (s_k, t) verändern.

⇒ damit ist die Aussage wahr.

a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9

5 2 1 3 1 4 0 3 4

$T[4]=2$ $T[6]$

⇝ 2 3 4 3 4

Bedeutung von $T[i]$

$T[i]$: Länge der längsten Teilfolge, die bei a_i endet.

$X_i := \{ j \in \mathbb{N} \mid 1 \leq j < i \text{ mit } a_i = a_j \pm 1 \}$

$T[i] = \max\limits_{j \in X_i} \{ T[j]+1, 1 \}$

⇝ Laufzeit $\sum\limits_{i=1}^{n} i \in \Theta(n^2)$

a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9

5 2 1 3 1 4 0 3 4

T: 1 1 2 2 3

1 0 1 0 1 3 2

1 2 3 4 5 1

$T[i]$: Länge der längsten Teilfolge, die nur Elemente aus $\{a_1, \dots, a_i\}$ verwendet

5 2 3 7 8 7 8 7

T: 1 1 2 2

$\sum\limits_{i=1}^{n} i$

$\underbrace{1+2+\dots+\overbrace{\tfrac{n}{2}+\dots+n}^{\frac{n}{2} \text{ Summanden}}}_{\geq \frac{n}{2}} \Rightarrow \geq \frac{n}{2} \cdot \frac{n}{2}$

$O(n^c)$ für Konstante c

(und Eingabegröße n)

$\Theta(n^3 \cdot \log n)$

$n^3 \cdot \log n \in O(n^4)$

$\in \Omega(n^3)$

$O(n^2)$: Algo ist nie langsamer als n^2

$\Omega(n^2)$: Es gibt Eingabe, bei der Algo $\geq n^2$ Schritte braucht

$\Theta(n^2)$: beides zusammen