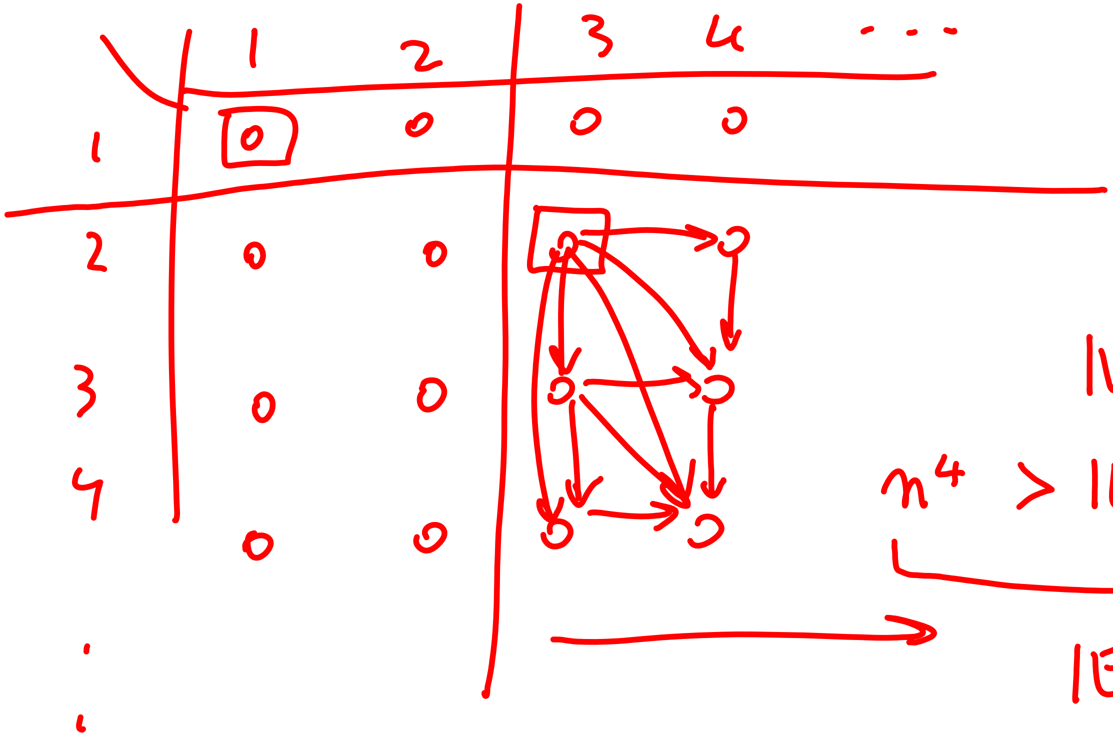
Dato $n \in \mathbb{N}$, si consideri il grafo orientato $G = (V, E)$, ove

$$
\begin{aligned}
V &= \{v_{ab} : 1 \leq a, b \leq n \ \ (a, b \in \mathbb{N})\} \\
E &= \{(v_{ab}, v_{cd}) : v_{ab}, v_{cd} \in V,\ v_{ab} \neq v_{cd},\ a \leq c,\ b \leq d\}\,.
\end{aligned}
$$

(a) Siano $w_1 : E \to \mathbb{R}$ e $w_2 : E \to \mathbb{R}^+$ due funzioni peso assegnate. Per ciascuno dei grafi pesati (G, w_1) e (G, w_2) si **illustri** un algoritmo per calcolare in maniera efficiente tutti i cammini minimi dal nodo v_{11}, **giustificando** opportunamente la scelta effettuata, e se ne **valuti** la complessità computazionale in funzione di $|V|$ e di $|E|$.

(b) Sapreste **esprimere** le complessità trovate nel punto precedente solo in funzione di n?



$w_1 : E \to \mathbb{R}$

$w_2 : E \to \mathbb{R}^+$

$|V| = n^2$

$$\underbrace{n^4 > |E| \geq |V| - 1 = n^2 - 1}$$

$|E| = O(n^3)$

$O(|E| + |V|) = O(n^3)$

ESERCIZIO 4 (esame completo/II prova in itinere)

Sia $G = (V, E)$ un grafo orientato con funzione peso $w : E \longrightarrow \mathbf{R}^+$ e sorgente $s \in V$, i cui nodi sono tutti raggiungibili da s.

(a) Si definiscano il *grafo* G'_s *dei cammini minimi da* s *in* G e la nozione di *albero dei cammini minimi da* s *in* G (rispetto alla funzione peso w).

(b) Dato un arco $(u, v) \in E$, si dimostri che (u, v) appartiene al grafo dei cammini minimi se e solo se $\delta(s, v) = \delta(s, u) + w(u, v)$, dove δ è la funzione distanza su G indotta da w.

(c) Si illustri un algoritmo efficiente per calcolare il grafo dei cammini minimi da s per il grafo pesato (G, w).

$(u,v) \in E'_s \longleftrightarrow \underbrace{\delta(s,v)}_{\text{minimo}} = \delta(s,u) + w(u,v)$

$(u,v) \in E'_s$ $\quad s \overset{\pi}{\rightsquigarrow} u \to v \rightsquigarrow \dots$

minimo

$\delta(s,v) = w(\pi;(u,v)) = w(\pi) + w(u,v) = \delta(s,u) + w(u,v)$

$(u,v) \in E : \delta(s,v) = \delta(s,u) + w(u,v)$

$s \overset{\pi}{\rightsquigarrow} u \to v$ $\qquad w(\pi) = \delta(s,u)$

$\pi' = \pi;(u,v)$

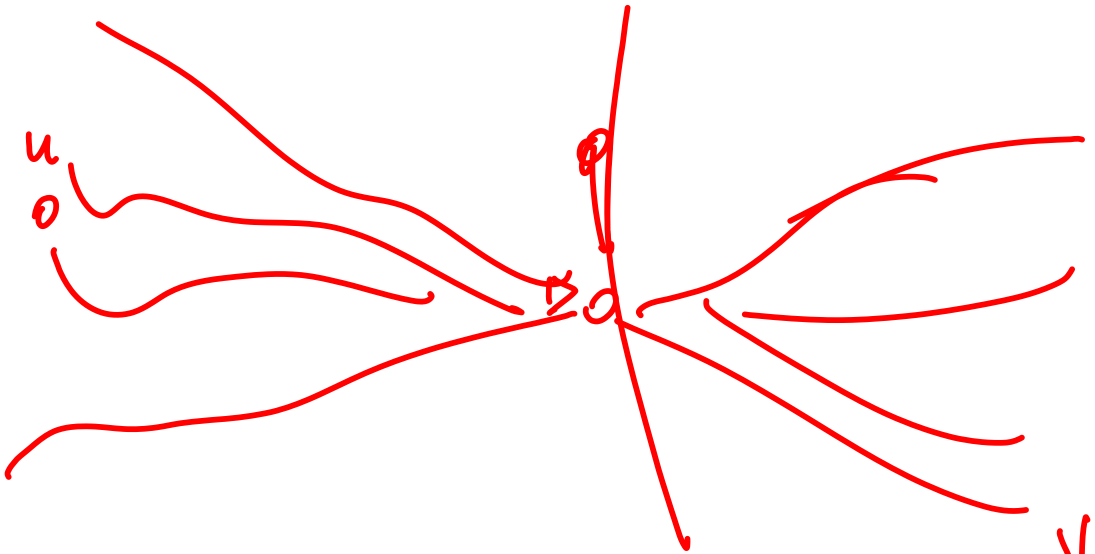
$w(\pi') = w(\pi) + w(u,v) = \delta(s,u) + w(u,v) = \delta(s,v) \longrightarrow (u,v) \in E'_s$

Sia $G = (V, E)$ un grafo orientato, $q \in V$ un nodo assegnato di G e $w : E \to \mathbb{R}^+$ una funzione peso in G a valori positivi.

Un *q-cammino* in G da u a v è un cammino in G da u a v passante per q.

Un q-cammino in G da u a v si dice *minimo* se il suo peso è minimo (relativamente alla funzione peso w) rispetto a tutti i q-cammini in G da u a v.

Si descriva un algoritmo efficiente che calcoli per ciascuna coppia di nodi $(u, v) \in V \times V$ un q-cammino minimo da u a v e se ne valuti la complessità computazionale.



$> O(V^3)$

$E + V \log V$

$V^2 (E + V \log V)$

$V^2 E + V^3 \log V$

In analogia con la nozione di *distanza* (minima) tra due nodi, si proponga una definizione della nozione di *distanza massima* tra due nodi in un grafo orientato pesato.

Quindi si descriva un algoritmo per determinare le distanze massime di tutti i nodi da una data sorgente $s \in V$ in un grafo orientato aciclico $G = (V, E)$ con funzione peso $w : E \longrightarrow \mathbf{R}$ e se ne valuti la complessità computazionale.

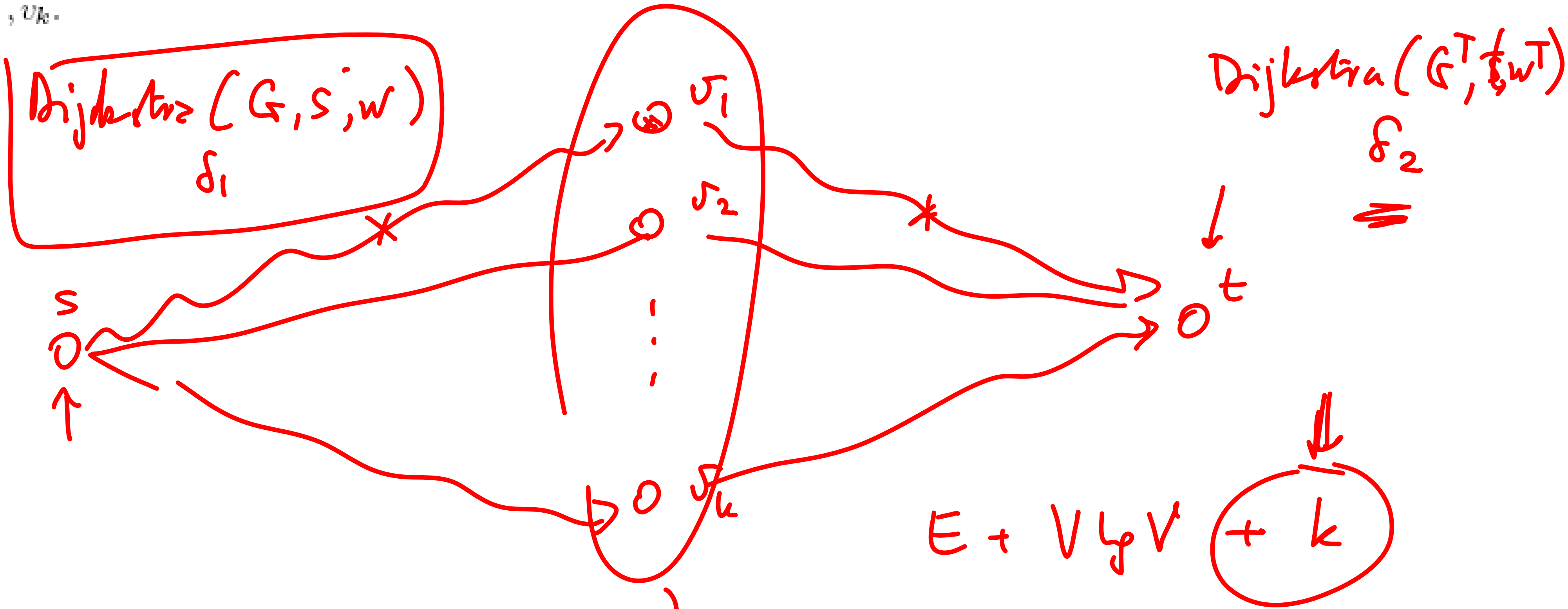
$$\Delta(s, v) = \sup \{ w(\pi) : \pi \in PATHS(G, s, v) \}$$

Sia $G = (V, E)$ un grafo orientato con una funzione peso non negativa $w : E \to \mathbb{R}^+$ e siano assegnati in G tre nodi distinti $s, q_1, q_2 \in V$.

Si descriva un algoritmo efficiente, valutandone anche la complessità computazionale, per calcolare tutti i cammini minimi (non necessariamente semplici) da s a ciascun nodo v di G, passanti per *almeno* uno dei due nodi assegnati q_1, q_2.

$$\delta(s, q_1) + \delta(q_1, v) < \delta(s, q_2) + \delta(q_2, v)$$

Sia $G = (V, E)$ un grafo orientato con una funzione peso non negativa $w : E \rightarrow \mathbb{R}^+$ e siano assegnati $(k + 2)$ nodi distinti $s, t, v_1, v_2, \ldots, v_k$ di G.
Si descriva un algoritmo efficiente, valutandone anche la complessità computazionale in funzione di $|V|$, $|E|$ e k, per calcolare un cammino minimo (non necessariamente semplice) da s a t, passante per *almeno* uno dei k nodi assegnati $v_1, v_2, \ldots, v_k$.

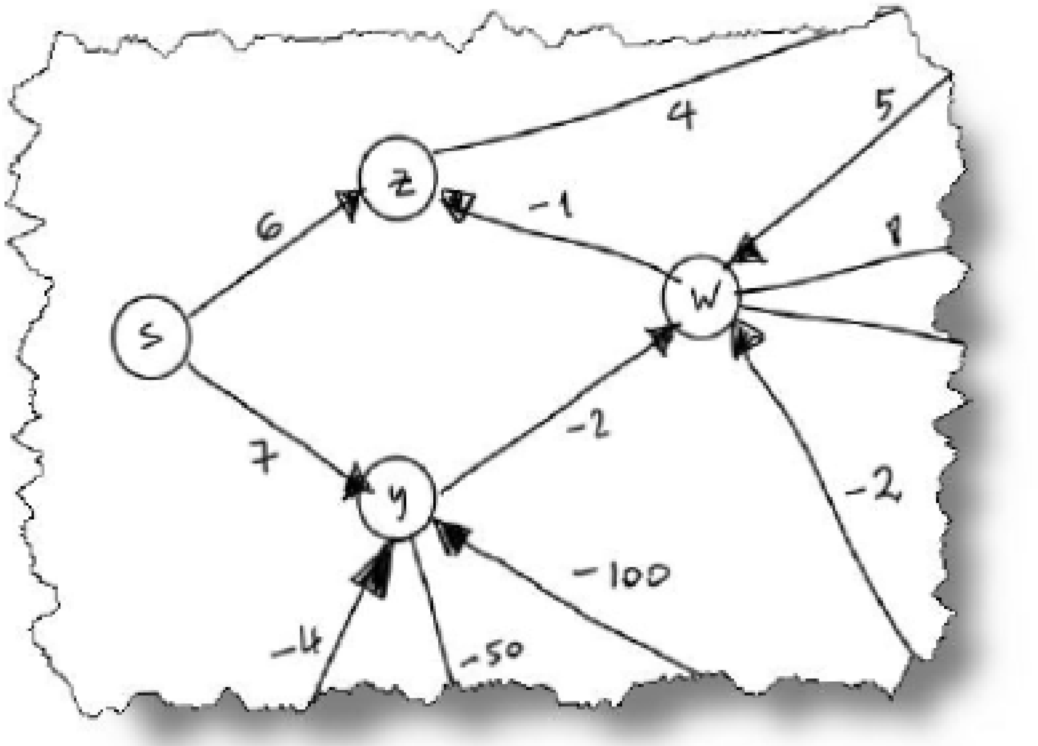


$$\min_{i=1\ldots k} \left(\underbrace{\delta_1(s, v_i) + \delta_2(v_i, t)} \right)$$

$w : E \rightarrow \mathbb{R}$

$w^T : E^T \rightarrow \mathbb{R}$

ESERCIZIO 1



Un importante problema di trasporto per l'azienda PIGA Petroli è stato ricondotto ad un problema di cammini minimi in un dato grafo orientato $G = (V, E)$, con funzione peso $w : E \to \mathbb{R}$, privo di cicli di peso negativo. In particolare, per effettuare le prime scelte strategiche, inizialmente serve solo calcolare la distanza in (G, w) dal nodo $s \in V$ al nodo $y \in V$. Di ciò è stato incaricato il brillante programmatore Turi N.G.[§] Questi preleva soltanto il frammento a lato dall'enorme rappresentazione grafica di (G, w) (si sappia infatti che $|V| = 10^8$) ed in men che non si dica calcola la distanza da s ad y e un cammino minimo da s ad y in (G, w).

Qual è il ragionamento effettuato dal nostro esperto programmatore Turi N.G.? *(Si applichino preferibilmente le proprietà sviluppate nel corso.)*
È possibile escludere con i dati in nostro possesso l'esistenza di altri cammini minimi da s ad y?
Che cosa si può dire della distanza di z da s?

[§] Per il rispetto della privacy sono state indicate solo le iniziali del cognome.

$$\delta(s, y) \le 7$$

$$s \xrightarrow{\alpha} z \xrightarrow{\pi'} y$$

$$w(\pi) = \alpha + w(\pi') < 7$$

$$w(\pi') < 7 - \alpha$$

$$\boxed{0 \le w(\pi') - 3 < 7 - \alpha - 3 = 4 - \alpha \longrightarrow \alpha \le 4}$$

Sia $G = (V, E)$ un grafo orientato con funzione peso $w : E \to \mathbb{R}^+$ e siano $s_1, s_2 \in V$ due nodi distinti di G.

Un cammino $\pi = (v_0, v_1, \ldots, v_k)$ in G si dice $\{s_1, s_2\}$-MINIMO se

$\underbrace{\delta_{\{s_1,s_2\}}(v_k)}$

- $v_0 \in \{s_1, s_2\}$
- per ogni cammino π' da s_1 a v_k o da s_2 a v_k si ha: $w(\pi) \leq w(\pi')$.

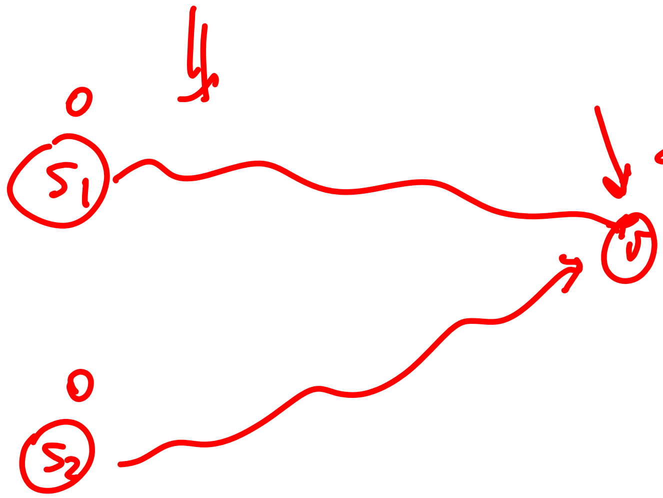
Inoltre, se $\pi = (v_0, v_1, \ldots, v_k)$ è un cammino $\{s_1, s_2\}$-minimo, poniamo $\delta_{\{s_1,s_2\}}(v_k) =_{Def} w(\pi)$.

Si illustri un algoritmo efficiente che per ciascun nodo v di G calcoli $\delta_{\{s_1,s_2\}}(v)$ nonché almeno un cammino $\{s_1, s_2\}$-minimo terminante in v.

Dijkstra$(G, \underline{s_1, s_2})$

$d[s] = 0$

$d[v] \geq \delta(s, v)$



$\min \begin{cases} \delta(s_1, v) \\ \delta(s_2, v) \end{cases} = \delta_{\{s_1,s_2\}}(v)$

$\delta_{\{s_1,s_2\}}(v) = \delta(s_1, v) \longrightarrow T_1$

$\longrightarrow T_2$