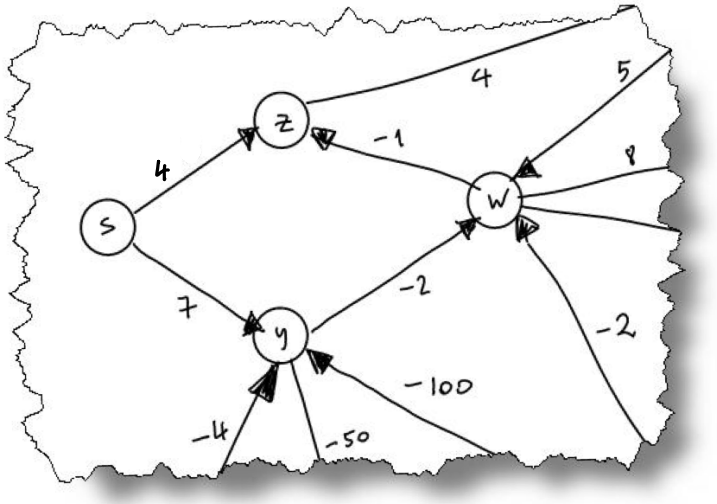
ESERCIZIO 1 FATTO



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.

ESERCIZIO 1 (Cammini minimi) FATTO

Sia (G, w), con $G = (V, E)$ e $w \colon E \to \mathbb{R}^+$, un grafo orientato pesato e siano $s_1, s_2 \in V$ due nodi distinti di V da cui è possibile raggiungere tutti i nodi in V.

Si descriva un algoritmo efficiente per trovare una partizione (V_1, V_2) di V tale che

$$V_1 \subseteq \{v \in V : \delta(s_1, v) \leq \delta(s_2, v)\} \qquad \text{e} \qquad V_2 \subseteq \{v \in V : \delta(s_2, v) \leq \delta(s_1, v)\}\,.$$

ESERCIZIO 2 (Cammini minimi) FATTO

Sia $G = (V, E)$ un grafo orientato con una funzione peso a valori reali $w : E \to \mathbb{R}$, ma senza cicli di peso negativo. Siano inoltre a, b, c tre nodi distinti di G.
Si progetti un algoritmo efficiente, valutandone anche la complessità computazionale, per determinare (qualora esista) un ciclo di peso minimo (non necessariamente semplice) passante per i tre nodi a, b, c, in un ordine qualsiasi.

ESERCIZIO 4 FATTO

Si descriva l'algoritmo di Johnson e lo si confronti dal punto di vista della complessità con l'algoritmo di Floyd-Warshall.

ESERCIZIO 2 FATTO

Si descriva l'algoritmo di Floyd-Warshall (anche mediante il suo pseudo-codice) e lo si confronti dal punto di vista della complessità con l'algoritmo di Johnson.

ESERCIZIO 3 FATTO

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.

La q-distanza da u a v in (G, w) è il peso del q-cammino da u a v di peso minimo (relativamente alla funzione w).

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

ESERCIZIO 3 FATTO

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 $w : E \longrightarrow \mathbb{R}$ e se ne valuti la complessità computazionale.

ESERCIZIO 2 (Cammini minimi) FATTO

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 definisca il *grafo* $G'_s = (V, E'_s)$ *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 G'_s 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 G'_s dei cammini minimi.

ESERCIZIO 3 (Cammini minimi) FATTO

Si illustri un algoritmo efficiente (anche mediante pseudo-codice) per determinare i cammini minimi da una sorgente assegnata a tutti i nodi da essa raggiungibili in un grafo orientato aciclico con funzione peso a valori reali.

ESERCIZIO 2 (Cammini minimi)

Si descriva l'algoritmo di Floyd-Warshall e il suo ambito di applicabilità. Quindi, dopo aver definito la nozione di chiusura transitiva di un grafo orientato, si descriva anche un algoritmo per calcolare la chiusura transitiva di un grafo orientato.

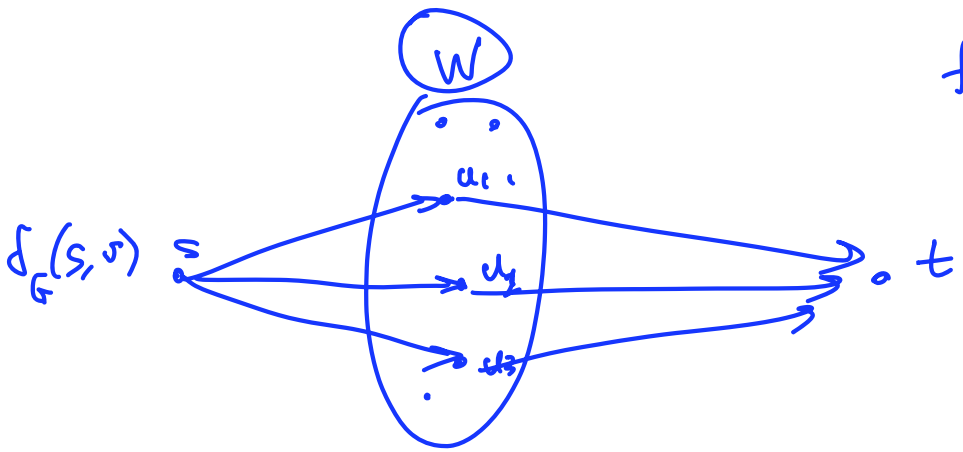
ESERCIZIO 3

Sia dato un grafo orientato $G = (V, E)$ con funzione peso $w : E \longrightarrow \mathbb{R}^+$ e siano s, t due nodi distinti assegnati di G tali che ogni nodo di G è raggiungibile da s e t è raggiungibile da ogni nodo di G.
Dato $\emptyset \neq W \subseteq V$, si ponga

$$f(W) =_{\text{Def}} \max\left\{\delta_G(u,t) : u \in W \textbf{ and } \delta_G(s,u) = \min_{v \in W} \delta_G(s,v)\right\},$$

dove δ_G è la funzione distanza in (G, w).

Si progetti un algoritmo per calcolare $f(W)$ e se ne valuti la complessità in funzione di $|V|$, $|E|$ e $|W|$.



δ_G := Dijkstra(G, w, s)

for $u \in W$ do

u_1 $\{u_2\}$

u_2 $\{u_1, u_2\}$

u_3 $\{u_3\}$

W'

Dijkstra(G^T, w^T, t)

for $u \in W'$ do

!

W''

Dijkstra $E + V \lg V$

$|W|$ $|V|$

$|W'|$ $|V|$

$E + V \lg V$

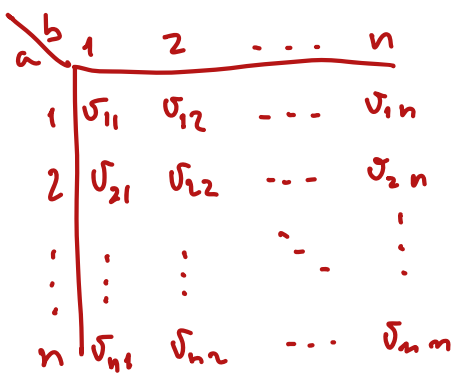
ESERCIZIO 4 (esame completo/II prova in itinere)

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

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

(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 ?

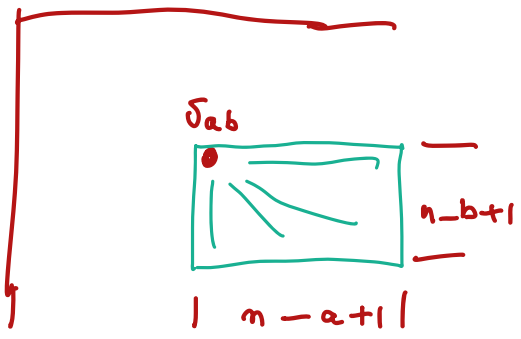


$E = \{(v_{ab}, v_{cd}) :$

$v_{ab} \neq v_{cd} \quad (a,b) = (c,d)$

$a \le c$

$b \le d$



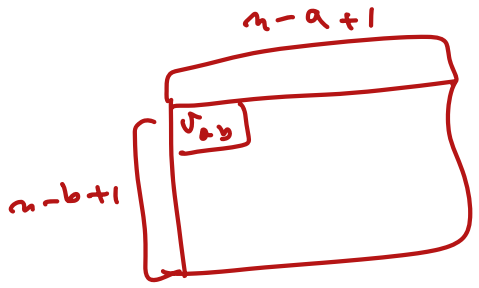
$w_1 : E \to \mathbb{R}$

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

(G, w_1), (G, w_2)

$|E| + |V| \log |V|$

$v_{ab} \longrightarrow v_{cd}$? $(n-a+1)(n-b+1)-1$



$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left[(n-i+1)(n-j+1)-1\right]$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}\left((n+1)^2-(n+1)i-(n+1)j+ij-1\right)$$

$$=n^2(n+1)^2-(n+1)\sum_{j=1}^{n}\sum_{i=1}^{n}i-(n+1)\sum_{i=1}^{n}\sum_{j=1}^{n}j+\sum_{i=1}^{n}\sum_{j=1}^{n}ij-n^2$$

$$=n^3(n+2)-\frac{n(n+1)^2}{2}-\frac{n(n+1)^2}{2}+\frac{n^2(n+1)^2}{4}$$

$$=n^3(n+2)-n(n+1)^2+\frac{n^2(n+1)^2}{4}$$

$$=\frac{4n^4+8n^3-4n^3-8n^2-4n+n^4+2n^3+n^2}{4}$$

$$=\frac{5n^4+6n^3-7n^2-4n}{4}$$

$$|E|=\frac{5n^4+6n^3-7n^2-4n}{4} \qquad |V|=n^2$$

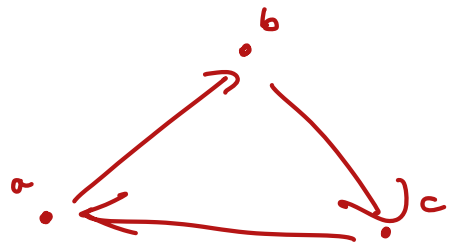
$$\sum_{i=1}^{n}\sum_{j=1}^{n}ij=\sum_{i=1}^{n}i\sum_{j=1}^{n}j=\sum_{i=1}^{n}\left(i\,\frac{n(n+1)}{2}\right)$$

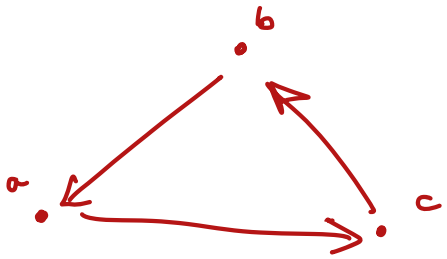
$$=\frac{n(n+1)}{2}\cdot\sum_{i=1}^{n}i$$

$$=\frac{n^2(n+1)^2}{4}$$

ESERCIZIO 2 (Cammini minimi)

Sia $G = (V, E)$ un grafo orientato con una funzione peso a valori reali $w : E \to \mathbb{R}$, ma senza cicli di peso negativo. Siano inoltre a, b, c tre nodi distinti di G.
Si progetti un algoritmo efficiente, valutandone anche la complessità computazionale, per determinare (qualora esista) un ciclo di peso minimo (non necessariamente semplice) passante per i tre nodi a, b, c, in un ordine qualsiasi.





BELLMAN-FORD (G, w, a)

BELLMAN-FORD (G, w, b)

BELLMAN-FORD (G, w, c)

if $\delta(a,b) + \delta(b,c) + \delta(c,a) < \delta(a,c) + \delta(c,b) + \delta(b,a)$ then

return $a \overset{min}{\rightsquigarrow} b \overset{min}{\rightsquigarrow} c \overset{min}{\rightsquigarrow} a$

else if $\delta(a,c) + \delta(c,b) + \delta(b,a) \neq +\infty$ then

return $a \overset{min}{\rightsquigarrow} c \overset{min}{\rightsquigarrow} b \overset{min}{\rightsquigarrow} a$

Complessità: $O(VE)$

ESERCIZIO 3

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.

La q-distanza da u a v in (G, w) è il peso del q-cammino da u a v di peso minimo (relativamente alla funzione w).

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

Dijkstra $(G, w, q) \Longrightarrow v \longmapsto \delta(q, v)$ $\quad O(E + V \lg V)$

Dijkstra $(G^T, w^T, q) \Longrightarrow v \longmapsto \delta^T(q, v)$ $\quad O(E + V \lg V)$

```
for u ∈ V & v ∈ V do
    δ_q(u,v) := δ^T(q,u) + δ(q,v)
return δ_q
```

$O(V^2)$

$O(V^2)$

ESERCIZIO 3

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 $w : E \longrightarrow \mathbb{R}$ e se ne valuti la complessità computazionale.

$$\Delta(s,u) = \sup_{\pi \in PATHS(G;s,u)} \delta(s,u)$$

Sia $w^- : E \to \mathbb{R}$ la funzione peso su E definita da

$$w^-(u,v) = -w(u,v)$$

$$(\delta^-, Pred^-) := DAG\text{-}SHORTEST\text{-}PATHS(G, w^-, s)$$

$$\Delta(s,u) = -\delta^-(s,u) \quad , \quad \forall u \in V$$

ESERCIZIO 2 (Cammini minimi)

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.

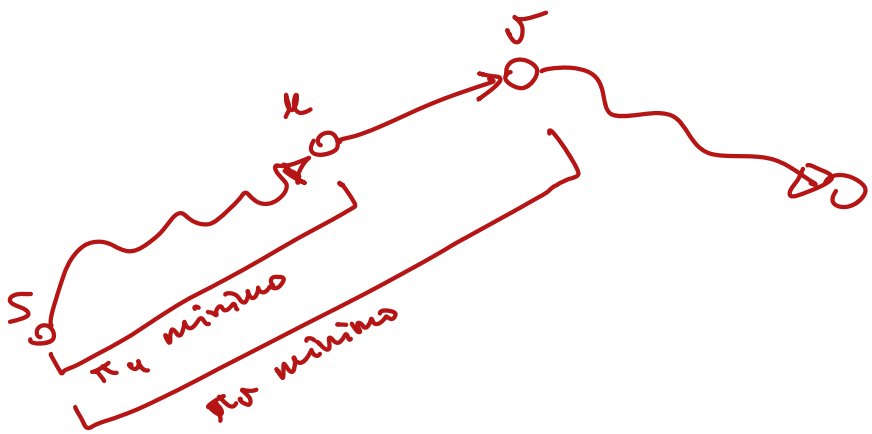
(a) Si definisca il *grafo* $G'_s = (V, E'_s)$ *dei cammini minimi da* s *in* G (rispetto alla funzione peso w).

(b) Dato un arco $(u, v) \in E$, si dimostri che (u, v) appartiene al grafo G'_s 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 G'_s dei cammini minimi.

(b) Sia $(u,v) \in E'_s$ GOAL: $\delta(s,v) = \delta(s,u) + w(u,v)$

$\longrightarrow \exists\ \pi$ minimo da s che contiene (u,v)



$\pi_v = \pi_u ; (u,v)$

$w(\pi_u) = \delta(s,u)$

$w(\pi_v) = \delta(s,v)$

$w(\pi_v) = w(\pi_u ; (u,v)) = w(\pi_u) + w(u,v)$

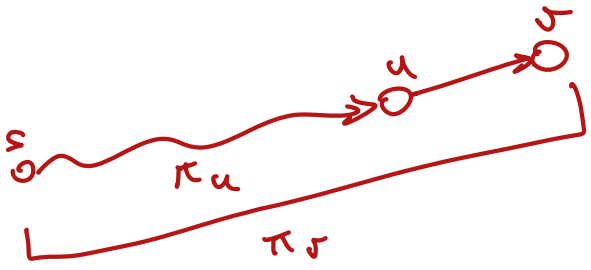
$\delta(s,v) = \delta(s,u) + w(u,v)$ ✓

Viceversa.

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

Sia π_u un cammino minimo da s a u, dunque di costo $\delta(s,u)$.

Sia $\pi_v = \pi_u ; (u,v)$



$$w(\pi_v) = w(\pi_u) + w(u,v)$$
$$= \delta(s,u) + w(u,v)$$
$$= \delta(s,v)$$

$\rightarrow$ π_v è un cammino minimo da s a v

$(u,v) \in \pi_v \longrightarrow (u,v) \in E'_s$.

$\delta := \text{DIJKSTRA}(G, w, s)$ — $O(E + V \lg V)$

$E'_s := \phi$

for $(u,v) \in E$ do — $O(E)$

  if $\delta(s,v) = \delta(s,u) + w(u,v)$ then

    $E'_s := E'_s \cup \{(u,v)\}$

return E'_s

$O(E + V \lg V)$