ALGORITMI PER IL CALCOLO DELLE DISTANZE E DEI CAMMINI MINIMI DA UNA DATA SORGENTE

- DESCRIVEREMO ALGORITMI BASATI SULLA TECNICA LABEL CORRECTING
- IN PRATICA, DATO UN GRAFO $G=(V,E)$, CON FUNZIONE PESO $w: E \to \mathbb{R}$ E SORGENTE s, VIENE MANTENUTA UNA FUNZIONE $d: V \to \mathbb{R} \cup \{+\infty\}$ (STIMA DELLA DISTANZA) IN MODO TALE CHE VALGA SEMPRE $d \geqslant \delta_{G_s}$, CIOE'
  $d[v] \geqslant \delta_{G_s}(v) = \delta_G(s,v)$, PER OGNI $v \in V$,
  E CHE ALLA FINE DELLA COMPUTAZIONE VALGA
  $d = \delta_{G_s}$, CIOE' $d[v] = \delta_{G_s}(v)$, PER OGNI $v \in V$.

INIZIALIZZAZIONE

- QUAL E' LA MIGLIORE STIMA DI d CHE POSSIAMO FARE INIZIALMENTE, PRIMA ANCORA DI CONSULTARE LA FUNZIONE w?
- E CHE COSA SI PUÒ DIRE INIZIALMENTE DELL' ALBERO DEI CAMMINI MINIMI ?

INIZIALIZZAZIONE

- QUAL E' LA MIGLIORE STIMA DI d CHE POSSIAMO FARE INIZIALMENTE, PRIMA ANCORA DI CONSULTARE LA FUNZIONE w?

  RISPOSTA

  $d[s] := 0$

  $d[v] := +\infty$, PER OGNI $v \in V \setminus \{s\}$

- OVVIAMENTE, IN TAL CASO VALE $d \geqslant \delta_{G_s}$

- L'ALBERO DEI CAMMINI MINIMI, RAPPRESENTATO IMPLICITAMENTE MEDIANTE UN ARRAY Pred, PUÒ ESSERE COSTRUITO CONTESTUALMENTE AL CALCOLO DELLA DISTANZA δ_{G_s}

  INIZIALMENTE BASTERÀ PORRE:

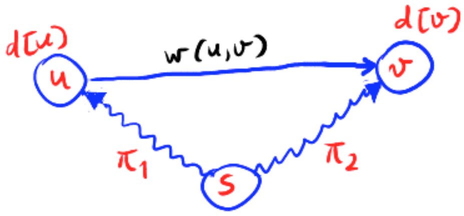
  $Pred[v] :=$ NIL, PER OGNI $v \in V$

```
procedure Initialize-Single-Source (G, s)
    for v ∈ V[G] do
        d[v] := +∞
        Pred[v] := NIL
    d[s] := 0
```

AGGIORNAMENTO DELLE FUNZIONI d E $Pred$

- SUPPONIAMO CHE $d \geq \delta_{G_s}$ E CHE PER OGNI NODO $v \in V$ TALE CHE $d[v] \neq +\infty$ LA FUNZIONE $Pred$ CONSENTA DI COSTRUIRE UN CAMMINO DA s A v DI PESO $\leq d[v]$.
- E' POSSIBILE MIGLIORARE LA STIMA DI d MANTENENDO LA PROPRIETA' DELLA FUNZIONE $Pred$?
- SE G AMMETTE CAMMINI MINIMI DA s LA RISPOSTA E' AFFERMATIVA.

- SI CONSIDERI LA SITUAZIONE



CON $w(\pi_1) \leq d[u]$ E $w(\pi_2) \leq d[v]$.

- SI OSSERVI CHE π_2 E $\pi_1' = \pi_1 ; (u,v)$ SONO DUE CAMMINI DISTINTI DA s A v, CON $w(\pi_1') \leq d[u] + w(u,v)$.
- QUINDI, SE $d[v] > d[u] + w(u,v)$ CONVERRA' PORRE:

  $d[v] := d[u] + w(u,v)$

  $Pred[v] := u$

NOTA: IL CAMMINO π_2 POTREBBE NON ESISTERE (SE $d[v] = +\infty$)

Procedure RELAX $(u,v;w)$

if $d[v] > d[u] + w(u,v)$ then

$d[v] := d[u] + w(u,v)$

$Pred[v] := u$

ALGORITMO GENERIC SINGLE-SOURCE SHORTEST-PATH

Procedure GSSSP(G,s,w)

Initialize-Single-Source (G,s)

while $\exists\ (u,v) \in E[G]$ TALE CHE $d[u]+w(u,v) < d[v]$ do

- sia (u,v) tale che $d[u]+w(u,v) < d[v]$

RELAX $(u,v;w)$

LEMMA DURANTE L'ESECUZIONE DI GSSSP(G,s,w), SI HA:

(a) LA FUNZIONE d DECRESCE MONOTONICAMENTE

(b) VALE SEMPRE $d \geq \delta_{G_s}$

(ANCHE IN ASSENZA DI CAMMINI MINIMI DA s)

DIM. LA (a) SEGUE IMMEDIATAMENTE PER INDUZIONE. PER QUANTO RIGUARDA LA (b), OSSERVIAMO CHE IMMEDIATAMENTE DOPO L'INIZIALIZZAZIONE VALE $d \geq \delta_{G_s}$. PER ASSURDO, SIA RELAX $(u,v;w)$ LA PRIMA CHIAMATA A RELAX DOPO LA QUALE $d \not\geq \delta_{G_s}$ (E QUINDI SI ABBIA $d[v] < \delta_{G_s}(v)$). SI HA:

- $+\infty > d[u] \geq \delta_{G_s}(u)$, PER CUI ESISTE UN CAMMINO π DA s A u IN G TALE CHE $w(\pi) \leq d[u]$
- $d[v] = d[u] + w(u,v) \geq w(\pi) + w(u,v) = w(\pi;(u,v)) \geq \delta_{G_s}(v)$, IN QUANTO $\pi;(u,v)$ E' UN CAMMINO DA s A v. ASSURDO. ■

- SIA $d : V \to \mathbb{R} \cup \{+\infty\}$ TALE CHE $d \geq \delta_{G_s}$ E $d[s] \leq 0$
- SI PONGA

$$\mathcal{P}_d = \{ \pi \in PATHS(G;s) : d[tail(\pi)] > w(\pi) \}$$

LEMMA $\mathcal{P}_d = \emptyset \longleftrightarrow (\forall (u,v) \in E)\ d[v] \leq d[u] + w(u,v)$

DIM ($\Rightarrow$) SIA $(u,v) \in E$ TALE CHE $d[v] > d[u] + w(u,v)$.
POICHE' $\delta_{G_s}(u) \leq d[u] < +\infty$, ESISTE $\pi_{su} \in PATHS(G;s,u)$
TALE CHE $w(\pi_{su}) \leq d[u]$.
SIA $\pi_{sv} = \pi_{su};(u,v) \in PATHS(G;s,v)$.
ALLORA $d[v] > d[u] + w(u,v) \geq w(\pi_{su}) + w(u,v) = w(\pi_{sv})$
E QUINDI $\pi_{sv} \in \mathcal{P}_d$, CIOE' $\mathcal{P}_d \neq \emptyset$. %

($\Leftarrow$) SE $\mathcal{P}_d \neq \emptyset$, SI SELEZIONI $\pi \in \mathcal{P}_d$ DI LUNGHEZZA MINIMA.
SI HA: $length(\pi) > 0$, IN QUANTO ALTRIMENTI SI AVREBBE $\pi = (s)$, $0 \geq d[s] > w(\pi) = 0$, ASSURDO.
SIA $\pi = (v_0, v_1, \ldots, v_k)$, CON $v_0 = s$ E $k \geq 1$. (QUINDI: $d[v_k] > w(\pi)$)
SI CONSIDERI $\pi' = (v_0, v_1, \ldots, v_{k-1})$.
PER LA MINIMALITA' DI $length(\pi)$, VALE $\pi' \notin \mathcal{P}_d$, DA CUI:
$d[v_{k-1}] \leq w(\pi')$,
PERTANTO: $d[v_k] > w(\pi) = w(\pi') + w(v_{k-1}, v_k) \geq d[v_{k-1}] + w(v_{k-1}, v_k)$.

■

LEMMA CONDIZIONE NECESSARIA E SUFFICIENTE PERCHE' L'ESECUZIONE DI GSSSP(G, s, w) TERMINI E' CHE $d = \delta_{G_s}$.

DIM (NECESSITA') SE GSSSP(G, s, w) SI FERMA, ALLORA VALE $(\forall (u,v) \in E)\ d[v] \leq d[u] + w(u,v)$, DA CUI PER IL LEMMA PRECEDENTE SI HA: $\mathcal{P}_d = \emptyset$.

SE PER ASSURDO $d \neq \delta_{G_s}$, ALLORA ESISTEREBBE $v \in V$ TALE CHE $d[v] > \delta_{G_s}(v)$ E QUINDI UN CAMMINO $\pi_{sv} \in PATHS(G; s, v)$ TALE CHE $d[v] > w(\pi_{sv})$. PERTANTO $\pi_{sv} \in \mathcal{P}_d \neq \emptyset$, ASSURDO.

(SUFFICIENZA) SE NEL CORSO DI UNA ESECUZIONE DI GSSSP(G,s,w) VALE $d = \delta_{G_s}$, ALLORA $\mathcal{P}_d = \emptyset$. INFATTI, SE PER ASSURDO ESISTESSE $\pi \in \mathcal{P}_d$, SI AVREBBE: $d[tail(\pi)] > w(\pi) \geq \delta_{G_s}(tail(\pi))$ E QUINDI $d \neq \delta_{G_s}$, ASSURDO. PER IL LEMMA PRECEDENTE SI HA LA TESI. ■

LEMMA SUPPONIAMO CHE (G, w) AMMETTA CAMMINI MINIMI DA UNA SORGENTE s.
SIANO $d : V \to \mathbb{R} \cup \{+\infty\}$ E $Pred : V \to V \cup \{NIL\}$ E SIA $T = (V_T, E_T)$ IL GRAFO INDOTTO DA $Pred$, CON $V_T = \{x \in V : d[x] \neq +\infty\}$ E $E_T = \{(Pred[x], x) : x \in V_T \setminus \{s\}\}$.
SUPPONIAMO CHE

(1) T SIA UN ALBERO RADICATO IN s

(2) PER OGNI $x, y \in V_T$ TALI CHE y SIA RAGGIUNGIBILE DA x IN T SI ABBIA $d[x] + w(T_{xy}) \leq d[y]$ (DOVE T_{xy} E' IL CAMMINO IN T DA x A y)

ALLORA, PER OGNI ARCO (u,v) DI G, DOPO L'ESECUZIONE DI RELAX$(u,v;w)$ LE PROPRIETA' (1) E (2) CONTINUANO A VALERE PER IL NUOVO GRAFO INDOTTO DA $Pred$.

DIM. DOPO L'ESECUZIONE DI RELAX(u,v;w), INDICHIAMO CON

- d' IL NUOVO VALORE DELLA FUNZIONE d
- $Pred'$ IL NUOVO VALORE DELLA FUNZIONE $Pred$

- SE $d[v] \leq d[u]+w(u,v)$, ALLORA: $d'=d$, $Pred'=Pred$ E QUINDI IL LEMMA E' BANALMENTE VERO.
- SUPPONIAMO CHE $d[v] > d[u]+w(u,v)$, COSICCHE' SI HA:
  - $d'[v] = d[u]+w(u,v)$ E $d'[z]=d[z]$, PER OGNI $z \neq v$
  - $Pred'[v]=u$ E $Pred'[z]=Pred[z]$, PER OGNI $z \neq v$
- SIA T' IL GRAFO INDOTTO DA $Pred'$.
- SIA T_{su} IL CAMMINO DA s AD u NELL'ALBERO T INDOTTO DA $Pred$.

./.

(1) SE T' NON FOSSE UN ALBERO RADICATO IN s, ALLORA v OCCORREREBBE NECESSARIAMENTE SU T_{su} E SI AVREBBE:

$$d[v] + w(T_{vu}) \leq d[u]$$
$$d[v] + w(T_{vu}) + w(u,v) \leq d[u] + w(u,v)$$
$$w(T_{vu}) + w(u,v) \leq d[u] + w(u,v) - d[v] < 0$$

CIOE' $T_{vu};(u,v)$ SAREBBE UN CICLO DI PESO NEGATIVO RAGGIUNGIBILE DA s E QUINDI (G,w) NON AMMETTEREBBE CAMMINI MINIMI DA s, ASSURDO.

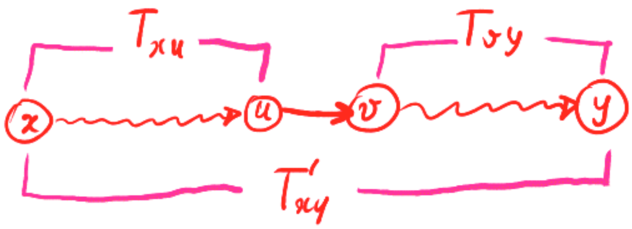
PERTANTO T' E' UN ALBERO RADICATO IN s.

./.

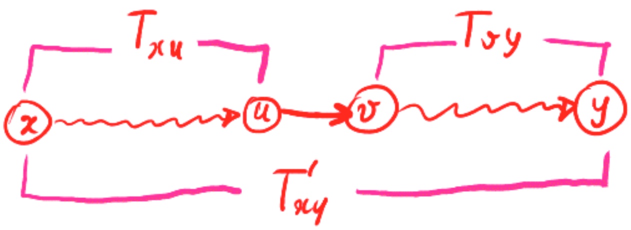
(2) E' SUFFICIENTE VERIFICARE LA PROPRIETA' (2) SOLTANTO PER TUTTE LE COPPIE $x, y \in V$ TALI CHE

- $x \neq v$
- v OCCORRE SUL CAMMINO T'_{xy} DA x A y IN T'

SIANO QUINDI T'_{xy}, T_{xu}, T_{vy} COME IN FIGURA



./.



SI HA:

$$
\begin{aligned}
d'[x] + w(T'_{xy}) &= d[x] + w(T_{xu}) + w(u,v) + w(T_{vy}) \\
&\leq d[u] + w(u,v) + w(T_{vy}) \\
&= d'[v] + w(T_{vy})
\end{aligned}
$$

- SE $y = v$, $d'[v] + w(T_{vy}) = d'[v] = d'[y]$,
- SE $y \neq v$, $d'[v] + w(T_{vy}) < d[v] + w(T_{vy}) \leq d[y] = d'[y]$.

IN OGNI CASO SI HA: $d'[v] + w(T_{vy}) \leq d'[y]$, E QUINDI:

$$d'[x] + w(T'_{xy}) \leq d'[y].$$ ∎

COROLLARIO NELLE IPOTESI CHE (G,w) AMMETTA CAMMINI MINIMI DA UNA DATA SORGENTE s, LE PROPRIETA' (1) E (2) DEL LEMMA PRECEDENTE SONO SODDISFATTE DURANTE OGNI ESECUZIONE DELLA PROCEDURA $GSSSP(G,s,w)$ (ED IN PARTICOLARE ALLA FINE, NEI CASI IN CUI $GSSSP(G,s,w)$ SIA TERMINANTE). ∎

COROLLARIO SE $GSSSP(G,s,w)$ TERMINA, $d=\delta_{G_s}$ E L'ALBERO T INDOTTO DALLA FUNZIONE $Pred$ E' UN ALBERO DI CAMMINI MINIMI IN (G,w) DA s.

DIM.

- SIA x UN NODO RAGGIUNGIBILE DA s IN G.
- SIA T_{sx} IL CAMMINO DA s A x NELL'ALBERO T
- PER LA (2) DEL LEMMA PRECEDENTE SI HA:

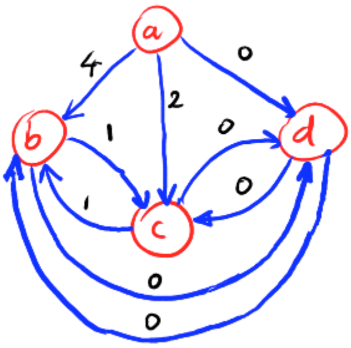
$$\delta_{G_s}(x) \leq w(T_{sx}) = d[s] + w(T_{sx}) \leq d[x] = \delta_{G_s}(x),$$

DA CUI $w(T_{sx}) = \delta_{G_s}(x)$,

CIOE' IL CAMMINO T_{sx} DA s A x E' MINIMO E QUINDI T E' UN ALBERO DI CAMMINI MINIMI. ∎

LEMMA SE (G,w) AMMETTE CAMMINI MINIMI DA s, ALLORA L'ESECUZIONE DI GSSSP(G,s,w) TERMINA DOPO AL PIÙ $e(V[G]-1)!$ PASSI. $\blacksquare$

ES. SI CONSIDERI IL SEGUENTE GRAFO:



RELAX$(a,b;w)$
RELAX$(b,c;w)$
RELAX$(c,d;w)$
RELAX$(b,d;w)$
RELAX$(d,c;w)$
RELAX$(a,c;w)$
RELAX$(c,b;w)$
RELAX$(b,d;w)$
RELAX$(c,d;w)$
RELAX$(d,b;w)$
RELAX$(a,d;w)$
RELAX$(d,b;w)$
RELAX$(b,c;w)$
RELAX$(d,c;w)$

OTTIMIZZAZIONI DI GSSSP

LEMMA SE $\pi : \overbrace{s \rightsquigarrow u}^{\pi_{su}} \to v$ E' UN CAMMINO MINIMO DA s A v IN (G, w) E AD UN CERTO PUNTO DI UN'ESECUZIONE DI $GSSSP(G, w, s)$ VALE $d[u] = \delta_{G_s}(u)$, ALLORA DOPO L'ESECUZIONE DI $RELAX(u, v; w)$ VARRA' $d[v] = \delta_{G_s}(v)$.

DIM. INFATTI DOPO LA CHIAMATA A $RELAX(u, v; w)$ SI AVRA':

$$\delta_{G_s}(v) \leq d[v] \leq d[u] + w(u,v) = \delta_{G_s}(u) + w(u,v) = w(\pi_{su}) + w(u,v)$$
$$= w(\pi) = \delta_{G_s}(v),$$

DA CUI $d[v] = \delta_{G_s}(v)$. ■

- PERCIO', SE SI ACCERTA CHE $d[u] = \delta_{G_s}(u)$, PER QUALCHE $u \in V[G]$, CONVERRA' CHIAMARE $RELAX(u, v; w)$ PER OGNI $v \in Adj_G[u]$.

```
Procedure SCAN(u; G, w)
    for v ∈ Adj_G[u] do
        RELAX(u, v; w)
```

• NELL' IPOTESI CHE ESISTANO CAMMINI MINIMI DA s IN (G,w) CONVERRA' UTILIZZARE LA SEGUENTE VARIANTE OTTIMIZZATA DI GSSSP.

Procedure OGSSSP(G,s,w)

Initialize-Single-Source(G,s)

$S := \emptyset$ // rappresenta l'insieme dei nodi x per cui e' stato
// accertato che vale $d[x] = \delta_{G_s}[x]$

while $\exists\, v \in V[G] \setminus S$ tale che $d[v] = \delta_{G_s}[v]$ **do**

- sia $v \in V[G] \setminus S$ tale che $d[v] = \delta_{G_s}[v]$
  SCAN$(v; G, w)$
  $S := S \cup \{v\}$

- PER LA CORRETTEZZA DI OGSSSP OCCORRE STABILIRE CHE AD OGNI ITERAZIONE DEL CICLO while VALGA:

$$V \setminus S \neq \emptyset \longrightarrow (\exists\, v \in V \setminus S)(d[v] = \delta_{G_s}(v))$$

- SIA $V \setminus S \neq \emptyset$ E SIA $u \in V \setminus S$.
  - SE $d[u] = \delta_{G_s}(u)$ ABBIAMO FINITO
  - SE $d[u] > \delta_{G_s}(u)$, SIA $\pi = (u_0, u_1, \ldots, u_k)$, CON $u_0 = s$ E $u_k = u$, UN CAMMINO MINIMO DA s AD u IN (G,w).
  - SIA $i_0 = \min_j u_j \in V \setminus S$, ALLORA: $d[u_{i_0}] = \delta_{G_s}(u_{i_0})$.

INFATTI:

CASO $i_0 = 0$: $d[u_{i_0}] = d[s] = 0 = \delta_{G_s}(s) = \delta_{G_s}(u_{i_0})$

CASO $i_0 > 0$: $u_{i_0 - 1} \in S$, QUINDI E' STATA ESEGUITA RELAX(u_{i_0-1}, u_{i_0}, w) CON $d[u_{i_0-1}] = \delta_{G_s}(u_{i_0-1})$, PER CUI DOPO SI HA: $d[u_{i_0}] = \delta_{G_s}(u_{i_0})$. ■

- LA COMPLESSITA' DELLA PROCEDURA OGSSSP E' $O(V+E)$ PIÙ IL TEMPO NECESSARIO A SELEZIONARE AD OGNI ITERAZIONE DEL CICLO while UN NODO $v \in V \setminus S$ TALE CHE $d[v] = d_{G_s}(v)$

- QUINDI PER OTTENERE DA OGSSSP ALGORITMI EFFETTIVI, OCCORRE TROVARE IL MODO PIÙ EFFICIENTE PER EFFETTUARE TALI SELEZIONI

- CONSIDEREREMO I SEGUENTI CASI:
  - CASO PIÙ GENERALE (ALGORITMO DI BELLMAN-FORD)
  - $w: E \to \mathbb{R}_0^+$ (ALGORITMO DI DIJKSTRA)
  - G ACICLICO

ALGORITMO DI BELLMAN-FORD

- LA STRATEGIA PIÙ SEMPLICE PER SELEZIONARE IL NODO GIUSTO CONSISTE NEL ... SELEZIONARE AD OGNI CICLO TUTTI I POSSIBILI NODI !

```
Procedure Bellman-Ford (G, s, w)
   Initialize-Single-Source (G, s)
   for i := 1 to |V[G]| do
       for (u,v) ∈ E[G] do
           RELAX (u,v;w)
   [
     ⋮
   ]
```

FUNZIONA NELL'IPOTESI CHE (G, w) AMMETTA CAMMINI MINIMI DA s.

? COME VERIFICARE SE (G, w) AMMETTE CAMMINI MINIMI DA s ?

ALGORITMO DI BELLMAN-FORD

- LA STRATEGIA PIÙ SEMPLICE PER SELEZIONARE IL NODO GIUSTO CONSISTE NEL ... SELEZIONARE AD OGNI CICLO TUTTI I POSSIBILI NODI !

```
Procedure Bellman-Ford (G, s, w)
    Initialize-Single-Source (G, s)
    for i := 1 to |V[G]| do
        for (u,v) ∈ E[G] do
            RELAX (u,v;w)
    for (u,v) ∈ E[G] do
        if d[v] > d[u] + w(u,v) then
            print ("cicli di peso negativo"); QUIT
```

COMPLESSITÀ:

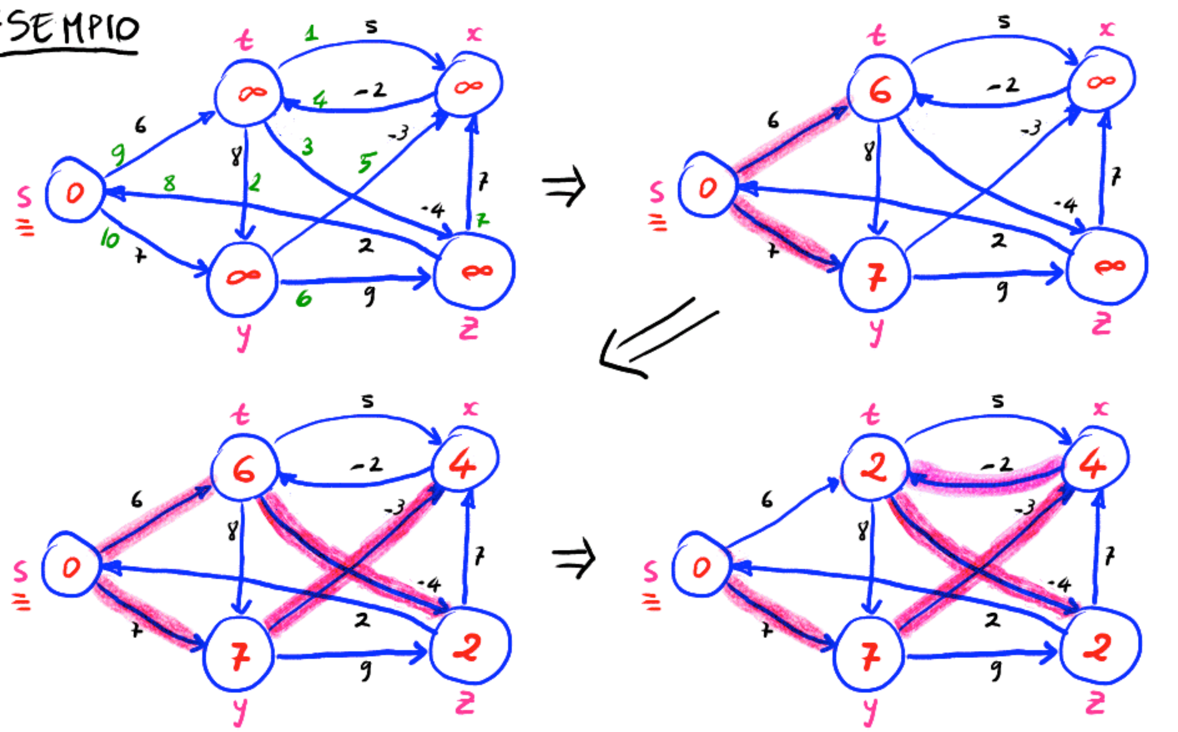
$O(V)$ +

$[O(V) \cdot O(E)]$

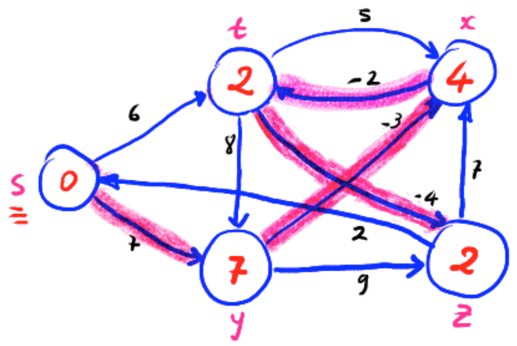
$O(E)$

$O(VE)$

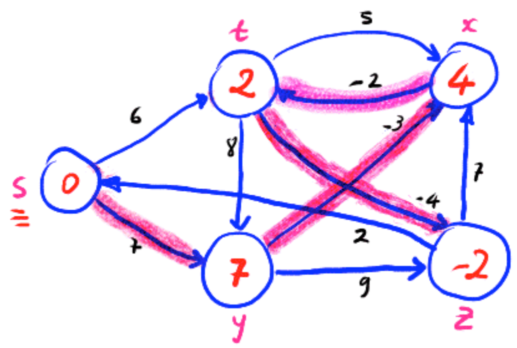
ESEMPIO



ESEMPIO (cntd)



$\Rightarrow$



ALGORITMO DI DIJKSTRA

- SUPPONIAMO CHE $w: E \to \mathbb{R}_0^+$ (CIOE' NON CI SONO ARCHI NEGATIVI)
- QUAL E' UN POSSIBILE CRITERIO EFFICIENTE PER SELEZIONARE $v \in V \setminus S$ TALE CHE $d[v] = \delta_{G_s}(v)$?

ALGORITMO DI DIJKSTRA

- SUPPONIAMO CHE $w: E \to \mathbb{R}_0^+$ (CIOE' NON CI SONO ARCHI NEGATIVI)
- QUAL E' UN POSSIBILE CRITERIO EFFICIENTE PER SELEZIONARE $v \in V \setminus S$ TALE CHE $d[v] = \delta_{G_s}(v)$?

PROPRIETA': SE $v \in V \setminus S$ E' TALE CHE $d[v] = \min \{ d[u] : u \in V \setminus S \}$ ALLORA $d[v] = \delta_{G_s}(v)$.

DIM. INFATTI, SE $d[v] > \delta_{G_s}(v)$, SIA $\pi = (v_0, v_1, \dots, v_k)$, CON $v_0 = s$, $v_k = v$ UN CAMMINO MINIMO IN (G, w) DA s A v. SIA $i_0 = \min_j v_j \in V \setminus S$. ALLORA $d[v_{i_0}] = \delta_{G_s}(v_{i_0})$.

QUINDI
$$d[v] > \delta_{G_s}(v) = w(\pi) = w(v_0, v_1, \dots, v_{i_0}) + w(v_{i_0}, \dots, v_k)$$
$$= \delta_{G_s}(v_{i_0}) + \sum_{j=i_0}^{k-1} w(v_j, v_{j+1}) \geq \delta_{G_s}(v_{i_0}) = d[v_{i_0}],$$
CONTRADDICENDO LA MINIMALITA' DI $d[v]$. ∎

ALGORITMO DI DIJKSTRA

Procedure Dijkstra (G, s, w)

  Initialize-Single-Source (G, s)

  $S := \emptyset$

  while $V[G] \setminus S \neq \emptyset$ do

  - sia $v \in V[G] \setminus S$ tale che $\boxed{d[v] = \min \{ d[u] : u \in V \setminus S \}}$

    SCAN$(v; G, w)$

    $S := S \cup \{v\}$

COME EFFETTUARE LA SELEZIONE DI v IN MANIERA EFFICIENTE?

ALGORITMO DI DIJKSTRA

Procedure Dijkstra (G, s, w)

    Initialize-Single-Source (G, s)
    Q := make-queue $(V[G], d)$
    while $Q \neq \emptyset$ **do**
        v := Extract-Min (Q, d)
        SCAN $(v; G, w)$

COMPLESSITA'

$O(V)$ +

$O(V)$ +

$[\,|V| \cdot$ costo(Extract_Min) +

$|E| \cdot$ costo(Decrease_Key)$]$

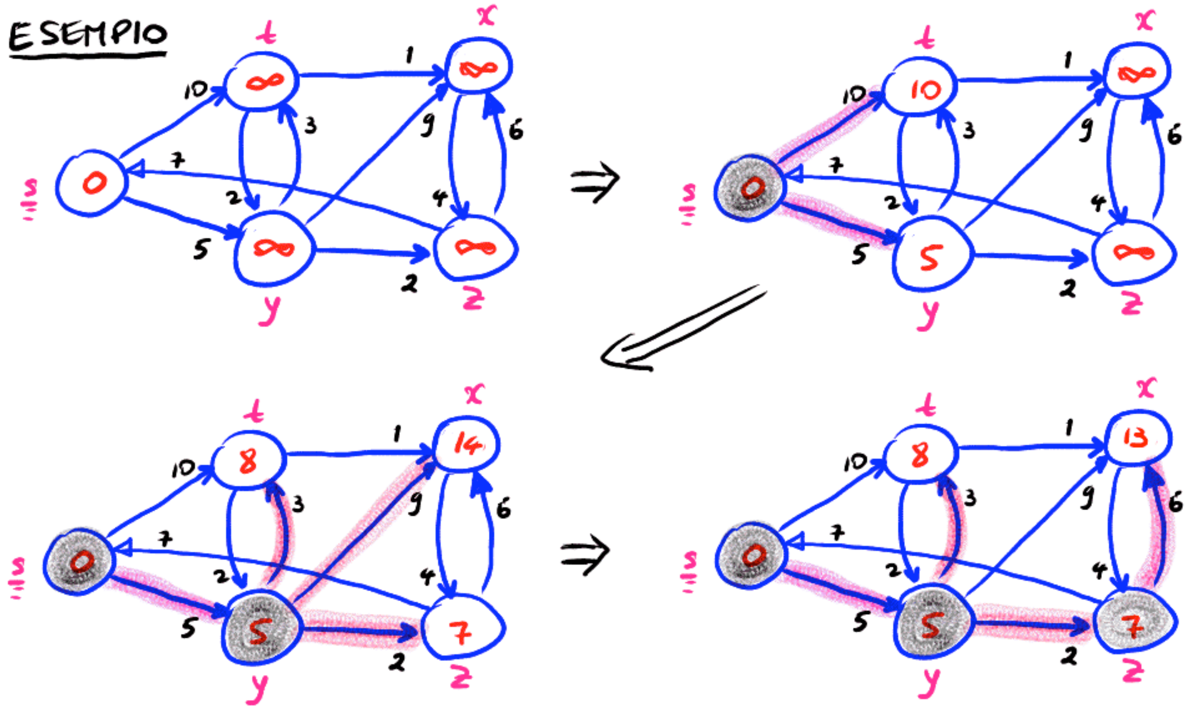
ALGORITMO DI DIJKSTRA

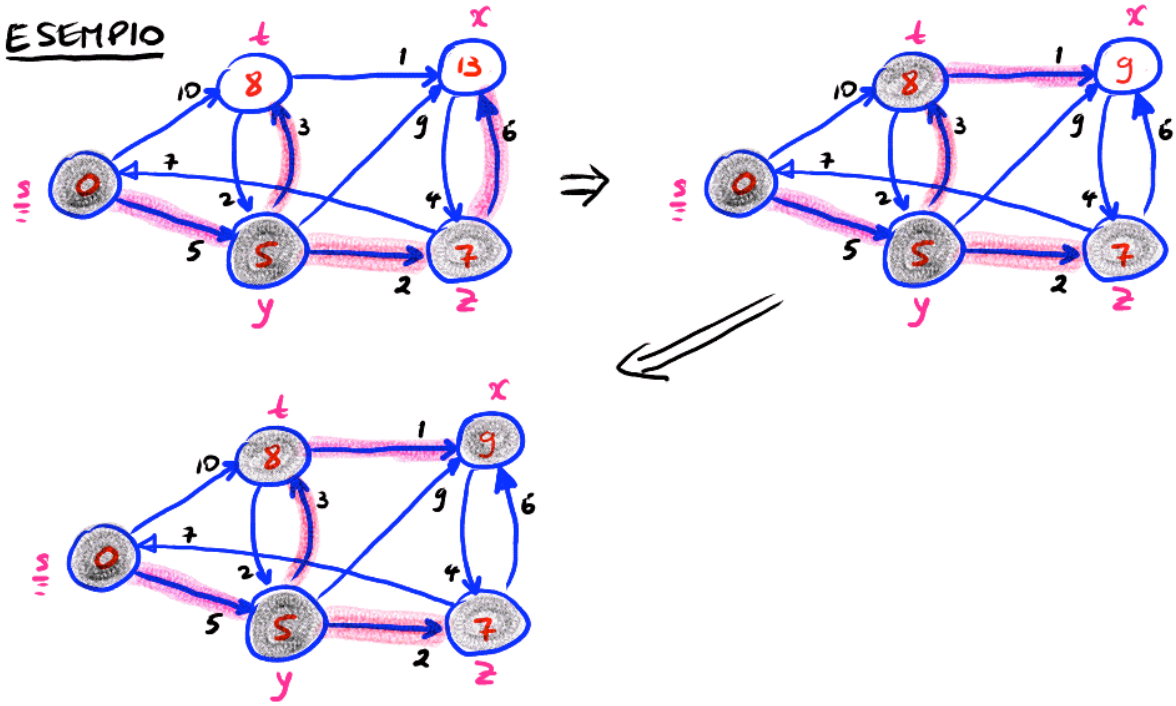
Procedure Dijkstra (G, s, w)

    Initialize-Single-Source (G, s)
    Q := make-queue $(V[G], d)$
    while $Q \neq \emptyset$ **do**
        v := Extract-Min (Q, d)
        SCAN $(v; G, w)$

DIVERSE IMPLEMENTAZIONI DELLA CODA Q

ARRAY	HEAP BINARIO	HEAP DI FIBONACCI
$O(V)$	$O(V)$	$O(V)$
$O(1)$	$O(V)$	$O(V)$
$O(V^2)$	$O(V \log V)$	$O(V \log V)$
$O(E)$	$O(E \log V)$	$O(E)$
$O(V^2)$	$O((V+E)\log V)$	$O(E+V\log V)$


ESEMPIO
s
t
x
y
z


ESEMPIO
s
t
x
y
z

CAMMINI MINIMI DA UNA SORGENTE IN GRAFI ACICLICI

- SUPPONIAMO CHE G SIA ACICLICO.
- QUAL E' UN POSSIBILE CRITERIO EFFICIENTE PER SELEZIONARE $v \in V \setminus S$ TALE CHE $d[v] = \delta_{G_S}(v)$?

CAMMINI MINIMI DA UNA SORGENTE IN GRAFI ACICLICI

- SUPPONIAMO CHE G SIA ACICLICO.
- QUAL E' UN POSSIBILE CRITERIO EFFICIENTE PER SELEZIONARE $v \in V \setminus S$ TALE CHE $d[v] = \delta_{G_S}(v)$?

PROPRIETA' SE TUTTI I PREDECESSORI IMMEDIATI DI $v \in V \setminus S$ SONO IN S, ALLORA $d[v] = \delta_{G_S}(v)$.

DIM. INFATTI, SE $d[v] > \delta_{G_S}(v)$, SIA $\pi = (v_0, v_1, \ldots, v_k)$, CON $v_0 = s$, $v_k = v$ UN CAMMINO MINIMO IN (G,w) DA s A v, SI HA $k \geq 1$, IN QUANTO $d[v_0] = d[s] = \delta_{G_S}(s) = 0$. POICHE' $v_{k-1} \in S$, $d[v_{k-1}] = \delta_{G_S}(v_{k-1})$. INOLTRE, SUBITO PRIMA DI INSERIRE v_{k-1} IN S VIENE ESEGUITA $RELAX(v_{k-1}, v_k; w)$, E QUINDI VALE $d[v] = d[v_k] = \delta_{G_S}(v_k) = \delta_{G_S}(v)$, ASSURDO. ■

CAMMINI MINIMI DA UNA SORGENTE IN GRAFI ACICLICI

Procedure DAG-Shortest-Paths'(G, s, w)

Initialize-Single-Source (G, s)
$S := \emptyset$
while $V[G] \setminus S \neq \emptyset$ **do**
  sia $v \in V[G] \setminus S$ tale che tutti i predecessori di v stiano in S
  SCAN$(v; G, w)$
  $S := S \cup \{v\}$

COME SELEZIONARE v IN MANIERA EFFICIENTE ?

CAMMINI MINIMI DA UNA SORGENTE IN GRAFI ACICLICI

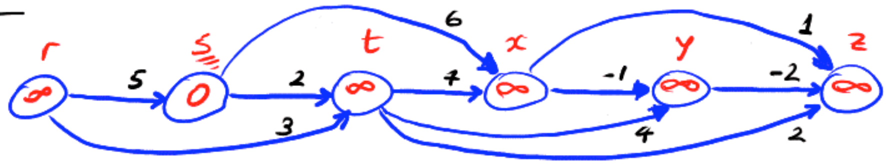
COMPLESSITA'

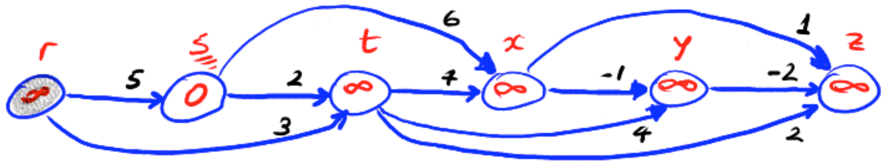
Procedure DAG-Shortest-Paths(G, s, w)

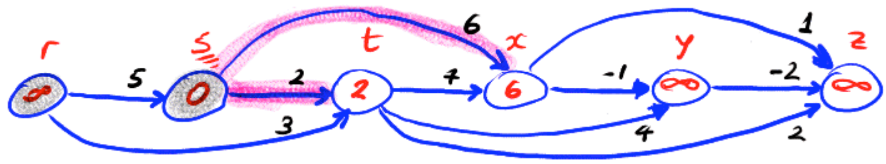
Initialize-Single-Source (G, s) — $O(V)$
sia $\prec$ un ordinamento topologico di G — $O(V+E)$
for $v \in V[G]$ seguendo l'ordinamento $\prec$ **do** — $O(V+E)$
  SCAN$(v; G, w)$
  $S := S \cup \{v\}$

$O(V+E)$

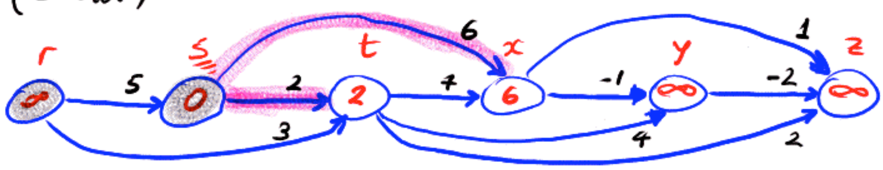
ESEMPIO

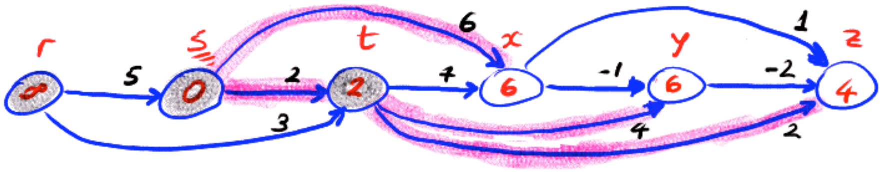


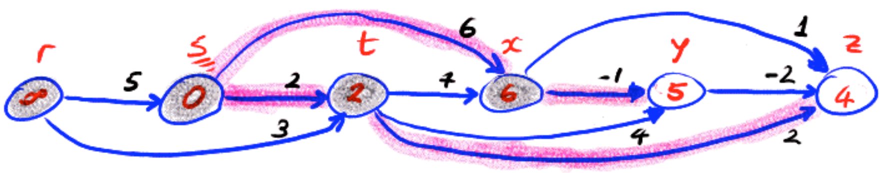




ESEMPIO (cntd)







ESEMPIO (cntd)