Algoritmi e Strutture Dati

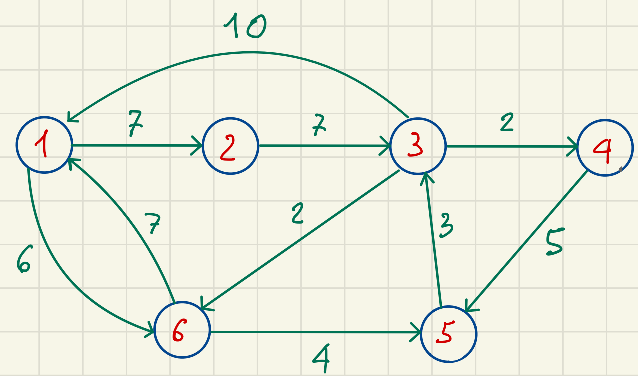
Lezione 25

19 novembre 2025

PROBLEMI "CAMMINI MINIMI"

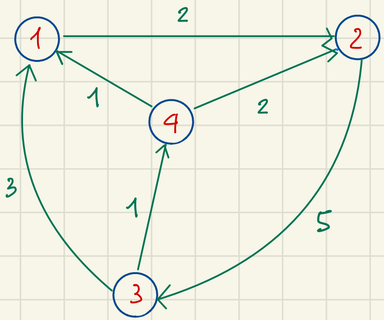
$G = (V, E) \quad w : E \to \mathbb{R}$

- Cammino minimo tra due vertici
- Cammini minimi da un vertice a tutti gli altri
- Cammini minimi tra ogni coppia di vertici



Cammini minimi tra tutte le coppie di vertici

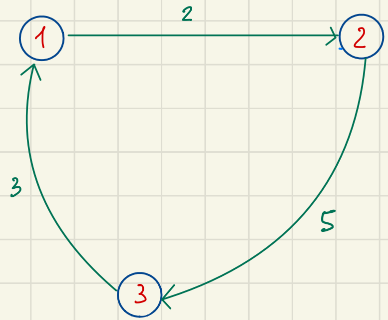
input



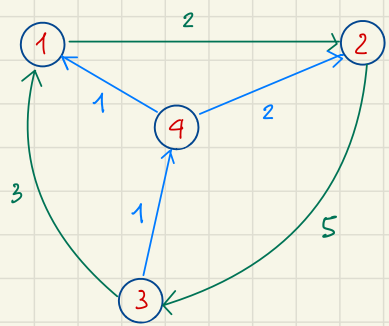
output

da \ a	1	2	3	4
1	0	2	7	8
2	7	0	5	6
3	2	3	0	1
4	1	2	7	0

matrice delle "distanze"



de \ a	1	2	3
1	0	2	7
2	8	0	5
3	3	5	0



$$2 \xrightarrow{6} 4 \xrightarrow{1} 1$$

de \ a	1	2	3	4
1	0	2	7	8
2	7	0	5	6
3	2	3	0	1
4	1	2	7	0

CAMMINI MINIMI TRA TUTTE LE COPPIE DI VERTICI:

L'ALGORITMO DI Floyd & Warshall

Supponiamo $V = \{v_1, v_2, \dots v_n\}$

d_{ij} = peso del cammino minimo da v_i a v_j

"lunghezza"

<u>Problema P</u>: determinare d_{ij} $i = 1, \dots, n$, $j = 1, \dots, n$

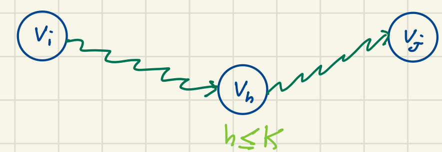
Problema $\mathcal{P}$: determinare d_{ij} $(i = 1, \dots, n, \quad j = 1, \dots, n)$

PROGRAMMAZIONE DINAMICA

Per $K = 0, \dots, n$ considero il problema vincolato:

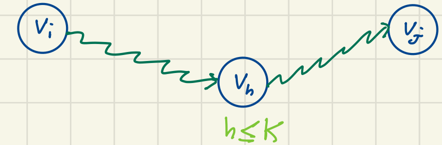
Problema $\mathcal{P}(K)$: determinare $d_{ij}^{(K)}$ $(i = 1, \dots, n, \quad j = 1, \dots, n)$

$d_{ij}^{(K)}$ = peso del cammino minimo da v_i a v_j che nei passi intermedi visita esclusivamente vertici di indice $\leq K$



Allora $\mathcal{P} = \mathcal{P}(n)$

Come risolvere $P(K)$?

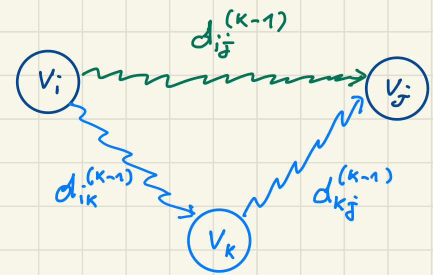


$\boxed{K=0}$ → vietati i vertici intermedi

$$d_{ij}^{(0)} = \begin{cases} w(v_i, v_j) & \text{se } \exists (v_i, v_j) \in E \text{ e } i \neq j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$

$\boxed{K>0}$

$$d_{ij}^{(K)} = \min \left\{ d_{ij}^{(K-1)},\ d_{iK}^{(K-1)} + d_{Kj}^{(K-1)} \right\}$$



$d_{iK}^{(K-1)} + d_{Kj}^{(K-1)} < d_{ij}^{(K-1)}$?

ALGORITMO FloydWarshall (Grafo G) → Matrice

Siano $D_0[1..n, 1..n], \dots, D_n[1..n, 1..n]$ matrici

```
FOR i ← 1 TO n DO
    FOR j ← 1 TO n DO
        IF i=j THEN D_0[i,j] ← 0
        ELSE IF (v_i, v_j) ∈ E THEN D_0[i,j] ← ω(v_i, v_j)
        ELSE D_0[i,j] ← ∞
```

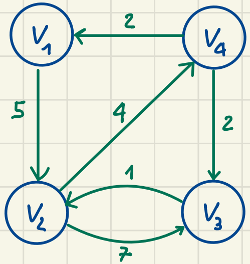
$$d_{ij}^{(0)} = \begin{cases} \omega(v_i, v_j) & \text{se } (v_i, v_j) \in E \text{ e } v_i \neq v_j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$

```
FOR k ← 1 TO n DO
    FOR i ← 1 TO n DO
        FOR j ← 1 TO n DO
            IF D_{k-1}[i,k] + D_{k-1}[k,j] < D_{k-1}[i,j] THEN
                D_k[i,j] ← D_{k-1}[i,k] + D_{k-1}[k,j]
            ELSE
                D_k[i,j] ← D_{k-1}[i,j]
RETURN D_n
```

$$d_{ij}^{(k)} = \min\left\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right\}$$

Tempo $O(n^3)$ Spazio $O(n^3)$

Esempio



$$M = \begin{pmatrix} \infty & 5 & \infty & \infty \\ \infty & \infty & 7 & 4 \\ \infty & 1 & \infty & \infty \\ 2 & \infty & 2 & \infty \end{pmatrix}$$

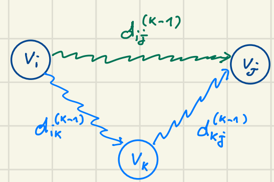
matrice dei pesi

$$D_0 = \begin{pmatrix} 0 & 5 & \infty & \infty \\ \infty & 0 & 7 & 4 \\ & 1 & 0 & \\ 2 & \infty & 2 & 0 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} \end{pmatrix} \qquad D_2 = \begin{pmatrix} \end{pmatrix} \qquad D_3 = \begin{pmatrix} \end{pmatrix}$$

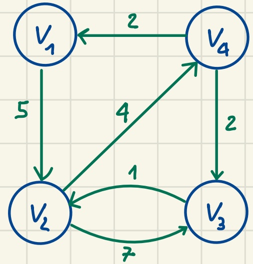
$$D_4 = \begin{pmatrix} \end{pmatrix}$$

$$d_{ij}^{(0)} = \begin{cases} w(v_i, v_j) & \text{se } (v_i, v_j) \in E \text{ e } v_i \neq v_j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$



$$d_{ij}^{(k)} = \min \{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \}$$

Esempio



$$M = \begin{pmatrix} \infty & 5 & \infty & \infty \\ \infty & \infty & 7 & 4 \\ \infty & 1 & \infty & \infty \\ 2 & \infty & 2 & \infty \end{pmatrix}$$

matrice dei pesi

$$D_0 = \begin{pmatrix} 0 & 5 & \infty & \infty \\ \infty & 0 & 7 & 4 \\ \infty & 1 & 0 & \infty \\ 2 & \infty & 2 & 0 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0 & 5 & \infty & \infty \\ \infty & 0 & 7 & 4 \\ \infty & 1 & 0 & \infty \\ 2 & 7 & 2 & 0 \end{pmatrix}$$

∞

$v_4 \rightsquigarrow v_2$

$2 \quad v_4 \rightsquigarrow v_1 \rightsquigarrow v_2 \quad 5$

$$D_2 = \begin{pmatrix} 0 & 5 & 12 & 9 \\ \infty & 0 & 7 & 4 \\ \infty & 1 & 0 & 5 \\ 2 & 7 & 2 & 0 \end{pmatrix}$$

∞

$v_1 \rightsquigarrow v_3$

$5 \quad v_1 \rightsquigarrow v_2 \rightsquigarrow v_3 \quad 7$

12

$$D_3 = \begin{pmatrix} 0 & 5 & 12 & 9 \\ \infty & 0 & 7 & 4 \\ \infty & 1 & 0 & 5 \\ 2 & 3 & 2 & 0 \end{pmatrix}$$

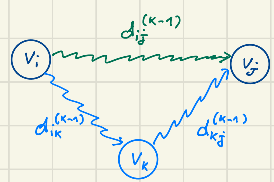
$$D_4 = \begin{pmatrix} 0 & 5 & 11 & 9 \\ 6 & 0 & 6 & 4 \\ 7 & 1 & 0 & 5 \\ 2 & 3 & 2 & 0 \end{pmatrix}$$

12

$v_1 \rightsquigarrow v_3$

$9 \quad v_1 \rightsquigarrow v_4 \rightsquigarrow v_3 \quad 2$

$$d_{ij}^{(0)} = \begin{cases} w(v_i, v_j) & \text{se } (v_i, v_j) \in E \text{ e } v_i \neq v_j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$



$$d_{ij}^{(k)} = \min \{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \}$$

$$d_{ij}^{(k)} = \min\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \}$$

• per $j = k$: $$d_{ik}^{(k)} = \min\{ d_{ik}^{(k-1)}, d_{ik}^{(k-1)} + \underbrace{d_{kk}^{(k-1)}}_{=0} \} = d_{ik}^{(k-1)}$$

• per $i = k$: $$d_{kj}^{(k)} = \min\{ d_{kj}^{(k-1)}, \underbrace{d_{kk}^{(k-1)}}_{=0} + d_{kj}^{(k-1)} \} = d_{kj}^{(k-1)}$$

IF $\overbrace{D_{k-1}[i,k]}^{D_k(i,k) =} + \overbrace{D_{k-1}[k,j]}^{D_k[k,j] =} < D_{k-1}[i,j]$ THEN

$D_k[i,j] \Leftarrow D_{k-1}[i,k] + D_{k-1}[k,j]$

ELSE

$D_k[i,j] \Leftarrow D_{k-1}[i,j]$

$\Longrightarrow$

IF $D[i,k] + D[k,j] < D[i,j]$

THEN

$D[i,j] \Leftarrow D[i,k] + D[k,j]$

ALGORITMO FloydWarshall (Grafo G) → Matrice

$\boxed{\text{Sia } D[1..n, 1..n] \text{ una matrice}}$

```
FOR i ← 1 TO n DO
    FOR j ← 1 TO n DO
        IF i = j THEN D[i,j] ← 0
        ELSE IF (v_i, v_j) ∈ E THEN D[i,j] ← ω(i,j)
        ELSE D[i,j] ← ∞
```

$D_0[i,j]$

$$d_{ij}^{(0)} = \begin{cases} \omega(v_i, v_j) & \text{se } (v_i, v_j) \in E \text{ e } v_i \neq v_j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$

```
FOR K ← 1 TO n DO
    FOR i ← 1 TO n DO
        FOR j ← 1 TO n DO
            IF D[i,K] + D[K,j] < D[i,j] THEN
                D[i,j] ← D[i,K] + D[K,j]
RETURN D
```

$$d_{ij}^{(K)} = \min\left\{ d_{ij}^{(K-1)},\ d_{iK}^{(K-1)} + d_{Kj}^{(K-1)} \right\}$$

```
IF D_{K-1}[i,K] + D_{K-1}[K,j] < D_{K-1}[i,j] THEN
    D_K[i,j] ← D_{K-1}[i,K] + D_{K-1}[K,j]
ELSE
    D_K[i,j] ← D_{K-1}[i,j]
```

D_n

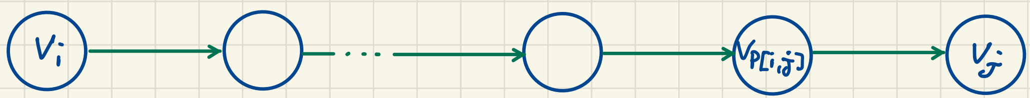
Tempo $O(n^3)$, spazio $O(n^2)$

Come ricavare il cammino minimo tra v_i e v_j?

Matrice ausiliaria P :

$P[i,j]$ = indice del penultimo vertice sul cammino minimo da v_i a v_j

(0 se $i=j$ o non c'è alcun cammino da v_i a v_j)



Dai valori contenuti alla fine in P si può ricavare il cammino

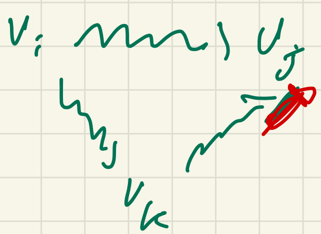
ALGORITMO FloydWarshall (Grafo G) → Matrice

Siano D[1..n, 1..n] una matrice , Sia P[1..n, 1..n] una matrice

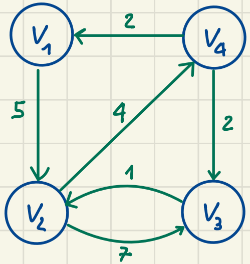
$$d_{ij}^{(0)} = \begin{cases} \omega(v_i, v_j) & \text{se } (v_i, v_j) \in E \text{ e } v_i \neq v_j \\ 0 & \text{se } v_i = v_j \\ \infty & \text{altrimenti} \end{cases}$$

```
FOR i ← 1 TO n DO
    FOR j ← 1 TO n DO
        P[i,j] ← 0
        IF i = j THEN D[i,j] ← 0
        ELSE IF (vi, vj) ∈ E THEN D[i,j] ← ω(i,j)
                                  P[i,j] ← i
        ELSE D[i,j] ← ∞

FOR K ← 1 TO n DO
    FOR i ← 1 TO n DO
        FOR j ← 1 TO n DO
            IF D[i,K] + D[K,j] < D[i,j] THEN
                D[i,j] ← D[i,K] + D[K,j]
                P[i,j] ← P[K,j]

RETURN D
```

$$d_{ij}^{(k)} = \min\{ d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \}$$



Esempio



$$M = \begin{pmatrix} \infty & 5 & \infty & \infty \\ \infty & \infty & 7 & 4 \\ \infty & 1 & \infty & \infty \\ 2 & \infty & 2 & \infty \end{pmatrix}$$

matrice dei pesi

$$D = \begin{pmatrix} 0 & 5 & 11 & 9 \\ 6 & 0 & 6 & 4 \\ 7 & 1 & 0 & 5 \\ 2 & 3 & 2 & 0 \end{pmatrix}$$

matrice delle distanze

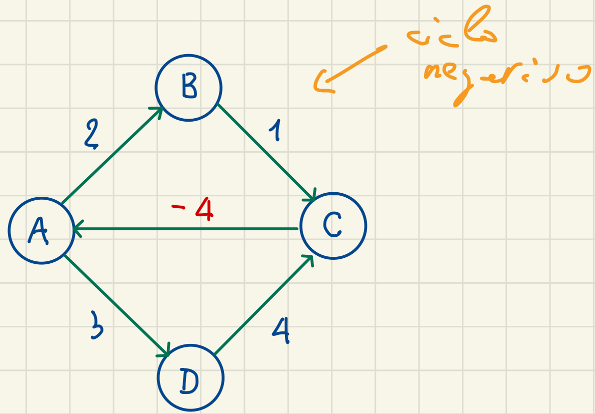
$$P = \begin{pmatrix} 0 & 1 & 4 & 2 \\ 4 & 0 & 4 & 2 \\ 4 & 3 & 0 & 2 \\ 4 & 3 & 4 & 0 \end{pmatrix}$$

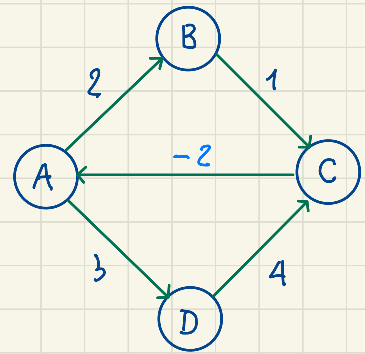
$$V_1 \to V_2 \to V_4 \to V_3$$

$$v_1 \overset{11}{\rightsquigarrow} v_3$$

$$v_1 \to v_2 \to v_4 \to v_3$$

PESI NEGATIVI ?

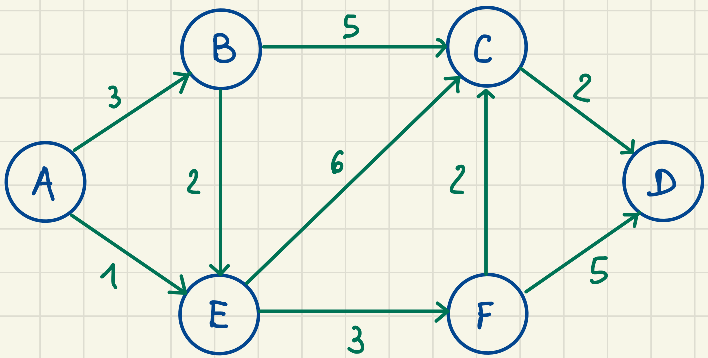




Floyd & Warshall : corretto anche con pesi negativi purché non ci siano cicli di peso negativi

cicli negativi → problema nel posto!

CAMMINI MINIMI DA UN VERTICE A TUTTI GLI ALTRI



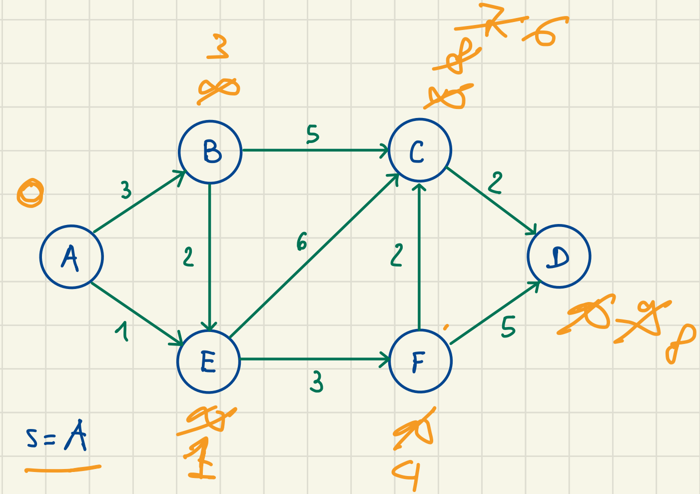
S = A

L'ALGORITMO DI Bellman & Ford

INPUT $G = (V, E)$ $\quad w: E \to \mathbb{R}$

$s \in V$ vertice di partenza

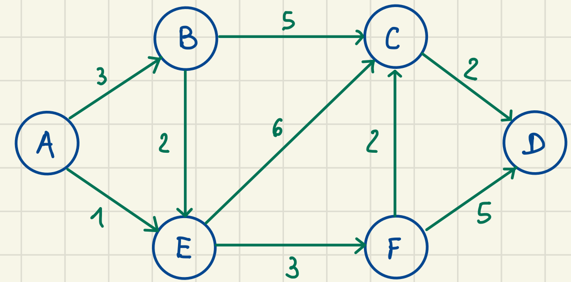
$d[v]$ = peso del cammino minimo da s a v sinora trovato



L'ALGORITMO DI Bellman & Ford

INPUT $G = (V, E)$ $\quad w : E \to \mathbb{R}$

$s \in V$ vertice di partenza

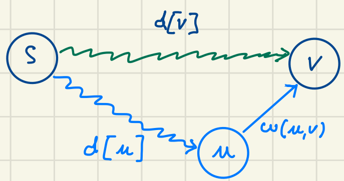


$s = A$

$d[v]$ = peso del cammino minimo da s a v sinora trovato

Inizialmente: $d[v] = \begin{cases} 0 & \text{se } v = s \\ \infty & \text{altrimenti} \end{cases}$

Aggiornamento:



$d[u] + w(u,v) < d[v]$?

Ad ogni passo si seguono TUTTI gli archi del grafo

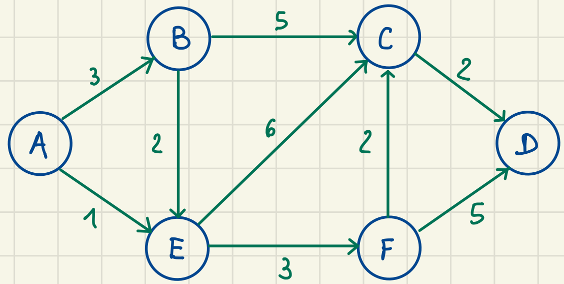
IF $d[u] + w(u,v) < d[v]$ THEN

$\quad d[v] \leftarrow d[u] + w(u,v)$

L'ALGORITMO DI Bellman & Ford

<u>INPUT</u> $G = (V, E)$ $\quad w: E \to \mathbb{R}$

$s \in V$ vertice di partenza



$s = A$

<u>Ipotesi</u> G privo di cicli negativi

$\Downarrow$

Tra ogni coppia di vertici esiste un cammino minimo semplice

$\Downarrow$

E' sufficiente considerare cammini composti da al più n vertici, e dunque $n-1$ archi

$\Rightarrow$ In al più $n-1$ passi si raggiungono tutti i nodi ottenendo i pesi dei cammini minimi

ALGORITMO Bellman & Ford (Grafo G, vertice s) $\rightarrow$ Vettore

Sia $d[V]$ un vettore con indici in V

$d[s] \leftarrow 0$

FOR EACH $v \in V \setminus \{s\}$ DO $d[v] \leftarrow \infty$

FOR $K \leftarrow 1$ TO $n-1$ DO

  FOR EACH $(u,v) \in E$ DO

    IF $d[u] + \omega(u,v) < d[v]$ THEN

      $d[v] \leftarrow d[u] + \omega(u,v)$

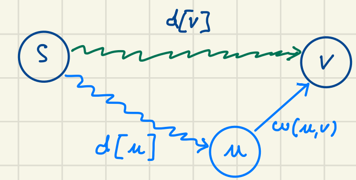
RETURN d

$d[v]$ = peso del cammino minimo da s a v sinora trovato

Inizialmente:

$$d[v] = \begin{cases} 0 & \text{se } v = s \\ \infty & \text{altrimenti} \end{cases}$$

Aggiornamento:



$d[u] + \omega(u,v) < d[v]$?

Lista di archi

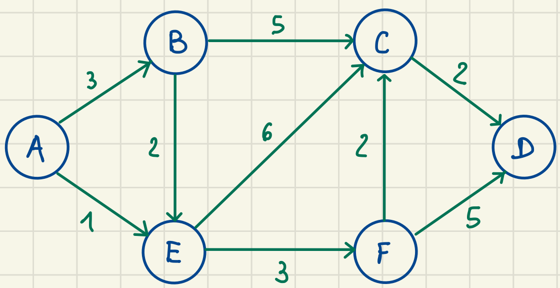
$m = \# E$

$n = \# V$

Tempo $O(m \cdot n)$

Spazio $O(n)$

L'ALGORITMO DI Dijkstra



INPUT $G = (V, E)$ $\quad w: E \to \mathbb{R}$

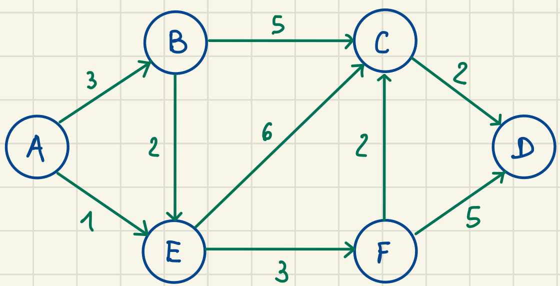
$s \in V$ vertice di partenza

Ipotesi: pesi non negativi

L'ALGORITMO DI Dijkstra

INPUT $G = (V, E)$ $\quad \omega: E \rightarrow \mathbb{R}$

$s \in V$ vertice di partenza



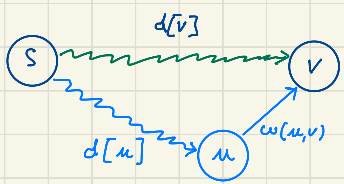
distanze "provvisorie"

$d[v]$ = peso del cammino minimo da s a v sinora trovato

Inizialmente:

$$d[v] = \begin{cases} 0 & \text{se } v = s \\ \infty & \text{altrimenti} \end{cases}$$

Aggiornamento:



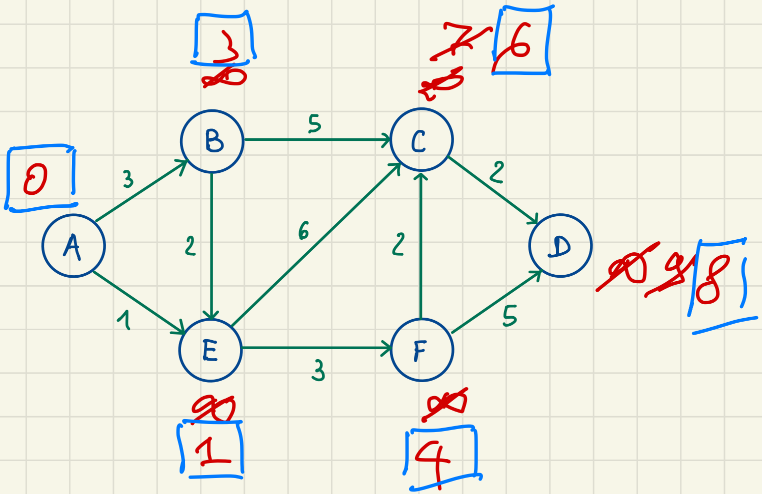
$d[u] + \omega(u,v) < d[v]$?

ad ogni passo si considerano gli archi che escono da un vertice u scelto con strategia GREEDY

IF $\quad d[u] + \omega(u,v) < d[v]$ THEN

$\quad d[v] \leftarrow d[u] + \omega(u,v)$

L'ALGORITMO DI Dijkstra



$s = A$

INPUT $G = (V, E)$ $\quad \omega : E \to \mathbb{R}$

$s \in V$ vertice di partenza

- Distanze provvisorie vettore $d[V]$

  inizialmente $d[v] = \begin{cases} 0 & \text{se } v = s \\ \infty & \text{se } v \neq s \end{cases}$

- $C \subseteq V$ insieme vertici candidati

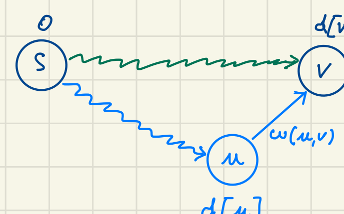
  inizialmente $C = V$

- strategia "greedy"

  preleva da C il vertice u con $d[u]$ minima

  $d[u]$ diventa definitiva

  aggiorna $d[v]$ per ogni v adiacente a u



$d[u] + \omega(u,v) < d[v]$?

ALGORITMO Dijkstra (Grafo G, vertice s) $\rightarrow$ Vettore

Sia $d[V]$ un vettore con indici in V

$d[s] \leftarrow 0$

FOR EACH $v \in V \setminus \{s\}$ DO $d[v] \leftarrow \infty$

$C \leftarrow V$

WHILE $C \neq \emptyset$ DO

  $u \leftarrow$ elemento di C con $d[u]$ minima

  $C \leftarrow C \setminus \{u\}$

  FOR EACH $(u,v) \in E$ DO

    IF $d[u] + \omega(u,v) < d[v]$ THEN

      $d[v] \leftarrow d[u] + \omega(u,v)$

RETURN d

- Distanze provvisorie vettore $d[V]$

  inizialmente $d[v] = \begin{cases} 0 & \text{se } v = s \\ \infty & \text{se } v \neq s \end{cases}$

- $C \subseteq V$ insieme vertici candidati

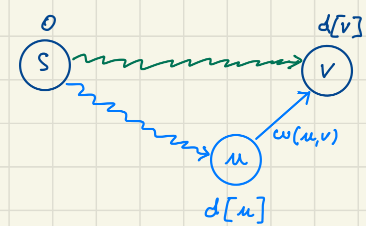
  inizialmente $C = V$

- strategia "greedy"

  preleva da C il vertice u con $d[u]$ minima

  $d[u]$ diventa definitiva

  aggiorna $d[v]$ per ogni v adiacente a u



$d[u] + \omega(u,v) < d[v]$?

ALGORITMO di DIJKSTRA: correttezza

Teo

Dato un grafo $G=(V,E,\omega)$ e un vertice $s \in V$, se G non contiene archi di peso negativo allora al termine dell'esecuzione dell'algoritmo di Dijkstra per ogni $v \in V$ $d[v]$ contiene la lunghezza del cammino minimo da s a v.

```
ALGORITMO Dijkstra (Grafo G, vertice s) → Vettore
  Sia d[V] un vettore con indici in V
  d[s] ← 0
  FOR EACH v ∈ V \ {s} DO d[v] ← ∞
  C ← V
  WHILE C ≠ ∅ DO
     u ← elemento di C con d[u] minima
     C ← C \ {u}
     FOR EACH (u,v) ∈ E DO
        IF d[u] + ω(u,v) < d[v] THEN
           d[v] ← d[u] + ω(u,v)
  RETURN d
```

ALGORITMO di DIJKSTRA: implementazione

lista di
adiacenza/incidenza

ALGORITMO Dijkstra (Grafo G, vertice s) $\to$ Vettore

Sia $d[V]$ un vettore con indici in V

$d[s] \leftarrow 0$

FOR EACH $v \in V \setminus \{s\}$ DO $d[v] \leftarrow \infty$

$C \leftarrow V$

WHILE $C \neq \emptyset$ DO

  $u \leftarrow$ elemento di C con $d[u]$ minima

  $C \leftarrow C \setminus \{u\}$

  FOR EACH $(u,v) \in E$ DO

    IF $d[u] + w(u,v) < d[v]$ THEN

      $d[v] \leftarrow d[u] + w(u,v)$

RETURN d