Algoritmi e Strutture Dati

Capitolo 13
Cammini minimi:
Algoritmo di Bellman e Ford

Cammini minimi in grafi:

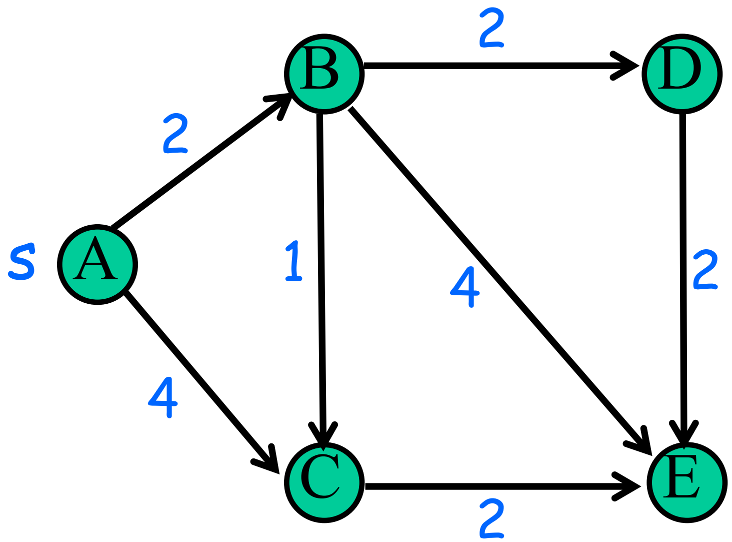
una trilogia

Cammini minimi in grafi:

Episodio II: cammini minimi a singola sorgente (per grafi con pesi negativi)

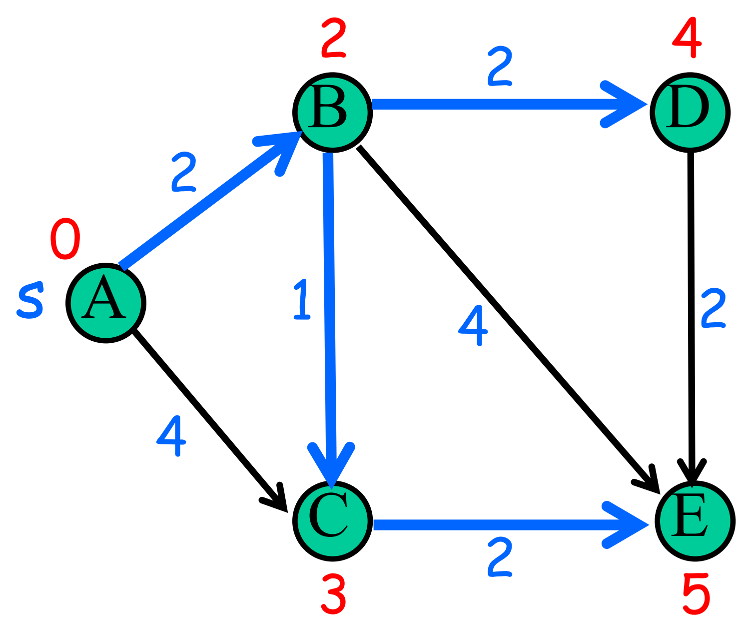
Problema del calcolo dei cammini minimi a singola sorgente

- Input:
  - grafo pesato $G=(V,E,w)$, $s \in V$
- Output:
  - albero dei cammini minimi radicato in s e/o le distanze di tutti i nodi da s, ovvero, $d_G(s,v)$ per ogni $v \in V$



Problema del calcolo dei cammini minimi a singola sorgente

- Input:
  - grafo pesato $G=(V,E,w)$, $s \in V$
- Output:
  - albero dei cammini minimi radicato in s e/o le distanze di tutti i nodi da s, ovvero, $d_G(s,v)$ per ogni $v \in V$



Nelle puntate precedenti:

- Visita BFS: albero dei cammini minimi per grafi non pesati. Tempo: $O(m+n)$.
- Algoritmo di Dijkstra: grafi con pesi non negativi. Tempo: $O(m+n \log n)$ con heap di Fibonacci. Altre implementazioni meno efficienti:
  - $O(n^2)$ - con array/liste non ordinate
  - $O(m \log n)$ con heap binari/binomiali

Algoritmo di Bellman e Ford

(albero dei cammini minimi in grafi che non contengono cicli negativi)

Distanza fra vertici

- Ricorda: la distanza d_{xy} tra due vertici x e y è il costo di un cammino minimo da x a y, o $+\infty$ se i due vertici non sono connessi
- Disuguaglianza triangolare: per ogni x, y e z $\in$ V

  $d_{xy} \leq d_{xz} + d_{zy}$ (l'uguaglianza sussiste quando esiste un cammino minimo da x a y che passa per z)
- Condizione di Bellman: per ogni arco (u,v) e per ogni vertice s, essendo $d_{uv} \leq w(u,v)$, dalla disuguaglianza triangolare segue che:

$$d_{sv} \leq d_{su} + d_{uv} \leq d_{su} + w(u,v)$$

Tecnica del rilassamento

- Partendo da stime per eccesso delle distanze $D_{xy} \geq d_{xy}$ si aggiornano le stime, decrementandole progressivamente fino a renderle esatte.

- Aggiornamento delle stime basato sul seguente passo di rilassamento (π_{vy} denota un qualche cammino tra v e y):

$$\text{(RILASSAMENTO)} \quad \textbf{if } (D_{xv} + w(\pi_{vy}) < D_{xy}) \quad \textbf{then } D_{xy} \leftarrow D_{xv} + w(\pi_{vy})$$

proprietà intuitive del rilassamento

- le stime iniziali sono upper bound alle distanze nel grafo e ogni sequenza di rilassamenti preserva questa invariante.
- le stime possono solo decrementare.
- quando una stima diventa esatta ed è quindi uguale alla distanza nel grafo, allora rilassare non ha alcun effetto.
- rilassare in generale non inficia mai le stime.

Ordine di rilassamento

Supponiamo di dover trovare un cammino minimo tra s e v_k. Sia π_{sv_k}=< s,v_1,v_2,… ,v_k > tale cammino. Osserviamo che π_{sv_k} potrebbe essere trovato eseguendo la seguente sequenza di rilassamenti:

$$
\begin{array}{llcllcl}
1. & D_{sv_1} & \leftarrow & D_{ss} & + & w(s, v_1) \\
2. & D_{sv_2} & \leftarrow & D_{sv_1} & + & w(v_1, v_2) \\
3. & D_{sv_3} & \leftarrow & D_{sv_2} & + & w(v_2, v_3) \\
\vdots & \vdots & & & & \\
k. & D_{sv_k} & \leftarrow & D_{sv_{k-1}} & + & w(v_{k-1}, v_k)
\end{array}
$$

Problema: Come faccio ad individuare la giusta sequenza di rilassamenti (ovviamente a priori non conosco la sequenza di archi di π_{sv_k})?

Approccio di Bellman e Ford

- Esegue n-1 passate
- In ciascuna passata, per ogni arco del grafo, esegui il relativo passo di rilassamento rispetto alla distanza dalla sorgente s
  - nota: alla fine della prima passata, con questo approccio esaustivo sono sicuro di eseguire anche il rilassamento $D_{sv_1} \leftarrow D_{ss} + w(s,v_1)$
- Dopo la j-esima passata, i primi j rilassamenti corretti sono stati certamente eseguiti (ovvero è stata trovata d_{sv_j} nonché la distanza tra s e tutti i nodi in G per i quali il cammino minimo da s è costituito da al più j archi)
- Alla fine della (n-1)-esima passata, ho trovato tutti i cammini minimi da s, poiché un cammino minimo contiene al più n-1 archi (il grafo non contiene cicli negativi e quindi esistono sempre cammini minimi semplici)

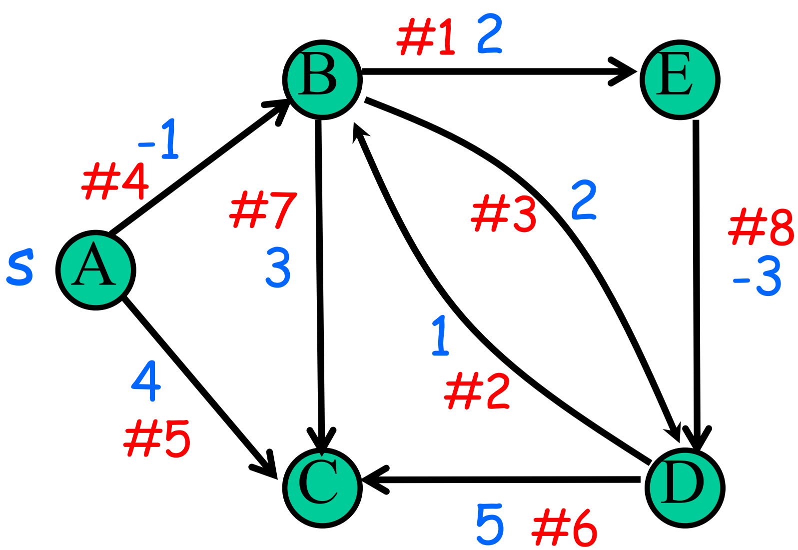
Pseudocodice

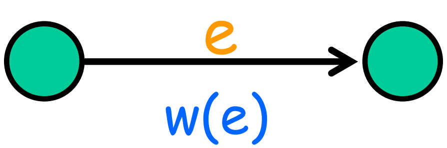
algoritmo `BellmanFord`$(grafo\ G, vertice\ s) \rightarrow distanze$
  inizializza D tale che $D_{sv} = +\infty$ per $v \neq s$, e $D_{ss} = 0$
  for $i = 1$ **to** $n-1$ **do**
    for each ($(u, v) \in E$) **do**
      if $(D_{su} + w(u, v) < D_{sv})$ **then** $D_{sv} \leftarrow D_{su} + w(u, v)$
  return D

Tempo di esecuzione: $\Theta(nm)$

(con liste di adiacenza)

applicare l'algoritmo di Bellman e Ford al seguente grafo





i: i-esimo arco considerato nel loop pricipale dell'alg

correttezza dell'algoritmo

assunzione: G non ha cicli di peso negativo

Lemma 1

> Dopo una sequenza qualsiasi di rilassamenti, vale $D_{sv} \geq d_{sv}$ per ogni $v \in V$

dim (per induzione sul # di rilassamenti)

all'inizio

$D_{ss}=0$ e $D_{sv}=+\infty$ per ogni v diverso da s

può essere $d_{ss}<0$? …no, altrimenti ci sarebbe un ciclo di peso negativo!

generico rilassamento

$$D_{sv} \leftarrow \underbrace{D_{su}}_{\geq d_{su}} + \underbrace{w(u,v)}_{\geq d_{uv}} \geq d_{su} + d_{uv} \geq d_{sv}$$

(per ipotesi induttiva)

per disug. triangolare

assunzione: G non ha cicli di peso negativo

Corollario 1

Se ad un certo punto vale $D_{sv}=d_{sv}$, allora la stima D_{sv} non sarà mai più cambiata

dim

Segue dal Lemma 1 e dal fatto che le stime possono solo decrescere

■

Teorema

> Se $G=(V,E)$ non ha cicli negativi l'algoritmo termina con $D_{sv}=d_{sv}$ per ogni $v \in V$

dim

poiché $D_{sv} \geq d_{sv}$ (Lemma 1) dobbiamo solo mostrare che a un certo punto vale $D_{sv}=d_{sv}$ (che poi non sarà più modificata per il Corollario 1)

sia $p=\langle s=v_0, v_1, \ldots, v_k=v \rangle$ un cammino minimo da s a v con il minimo numero di archi

$\Rightarrow$ p è semplice (attenzione a cicli di costo 0!) $\Rightarrow$ $k \leq n-1$

dimostro per induzione che dopo i passate dell'alg, vale $D_{sv_i} = d_{sv_i}$

caso base: $D_{ss}=0 = d_{ss}$ perché non ci sono cicli negativi

caso induttivo: quando rilasso arco (v_{i-1}, v_i)

$$D_{sv_i} = D_{sv_{i-1}} + w(v_{i-1}, v_i) = d_{sv_{i-1}} + w(v_{i-1}, v_i) = d_{sv_i}$$

(per ipotesi induttiva) per sottostruttura ottima e se il grafo ha dei cicli negativi (raggiungilibi da s)?

…posso modificare l'algoritmo di Bellman e Ford per verificarne l'esistenza!

Pseudocodice

algoritmo BellmanFord$(grafo\ G, vertice\ s) \rightarrow distanze$
  inizializza D tale che $D_{sv} = +\infty$ per $v \neq s$, e $D_{ss} = 0$
  for $i = 1$ **to** $n-1$ **do**
    for each $((u,v) \in E)$ **do**
      if $(D_{su} + w(u,v) < D_{sv})$ **then** $D_{sv} \leftarrow D_{su} + w(u,v)$
  for each $(u,v) \in E$ **do**
    if $(D_{su}+w(u,v)<D_{sv})$ **then return** "esiste ciclo negativo"
  return D

Tempo di esecuzione: $\Theta(n\,m)$

(con liste di adiacenza)

Correttezza algoritmo

se G non ha cicli negativi, nell'ultimo ciclo **for** ➡ la condizione dell'**if** non è mai soddistatta

ma se G ha un ciclo negativo (raggiungibile da s) l'algoritmo se ne accorge?

sia <$v_1, v_2, \dots, v_{k-1}, v_k = v_1$> un ciclo di peso negativo (raggiungibile da s)

supponiamo per assurdo che l'algoritmo non se ne accorge

condizione dell'if dell'ultimo ciclo for è sempre falsa, ovvero:

$$D_{sv_i} + w(v_i, v_{i+1}) \geq D_{sv_{i+1}} \qquad i=1,2,\dots,k-1$$

sommando su tutti gli indici i:

$$\sum_{i=1}^{k-1} D_{sv_i} + \sum_{i=1}^{k-1} w(v_i, v_{i+1}) \geq \sum_{i=1}^{k-1} D_{sv_{i+1}}$$

i termini delle due sommatorie sono uguali

$$\Rightarrow \sum_{i=1}^{k-1} w(v_i, v_{i+1}) \geq 0$$

ma il ciclo era negativo: assurdo!

- **osservazione**:
  - se conoscessi l'ordine in cui rilassare gli archi, potrei trovare le distanze dalla sorgente facendo una sola passata di rilassamenti

- **domanda**: posso conoscere questo ordine?

  - …a volte sì: per esempio se G è un DAG!

Algoritmo basato su ordinamento topologico

(albero dei cammini minimi in grafi aciclici)

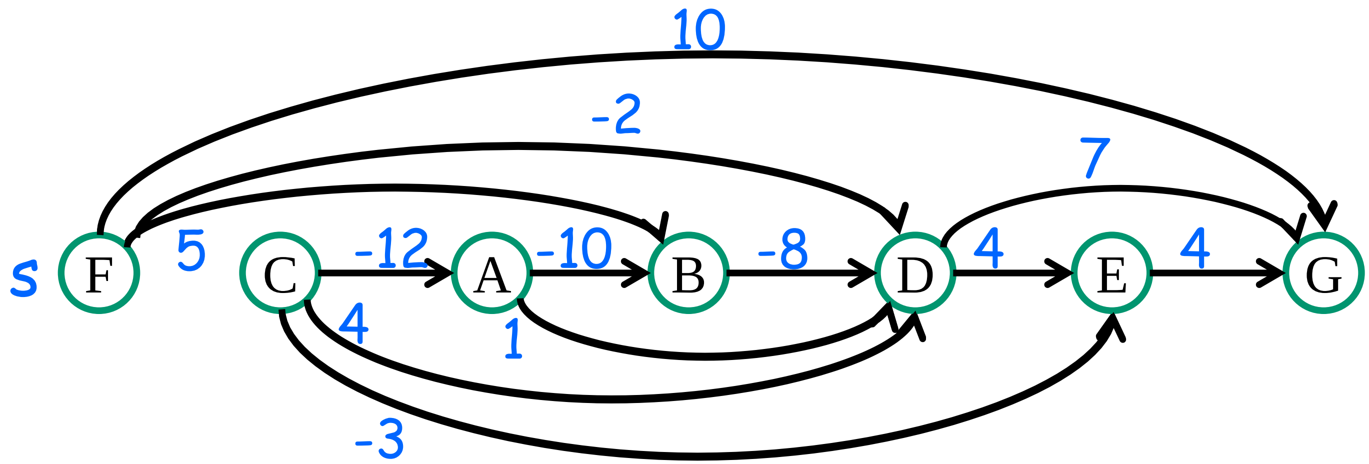
Cammini minimi in grafi aciclici

Eseguo i rilassamenti in ordine topologico, da sinistra verso destra: infatti, poiché tutti gli archi sono orientati verso destra, le stime di distanza che mi lascio alle spalle sono esatte (non possono essere ulteriormente rilassate)!

algoritmo `distanzeAciclico`$(grafo\; G, vertice\; s) \rightarrow distanze$
  inizializza D tale che $D_{sv} = +\infty$ per $v \neq s$, e $D_{ss} = 0$
  $ord \leftarrow$ `ordinamentoTopologico`(G)
  for $i = 1$ **to** n **do**
    sia u l'i-esimo vertice nell'ordinamento topologico ord
    for each $(\,(u, v) \in E\,)$ **do**
      if $(D_{su} + w(u, v) < D_{sv})$ **then** $D_{sv} \leftarrow D_{su} + w(u, v)$
  return D

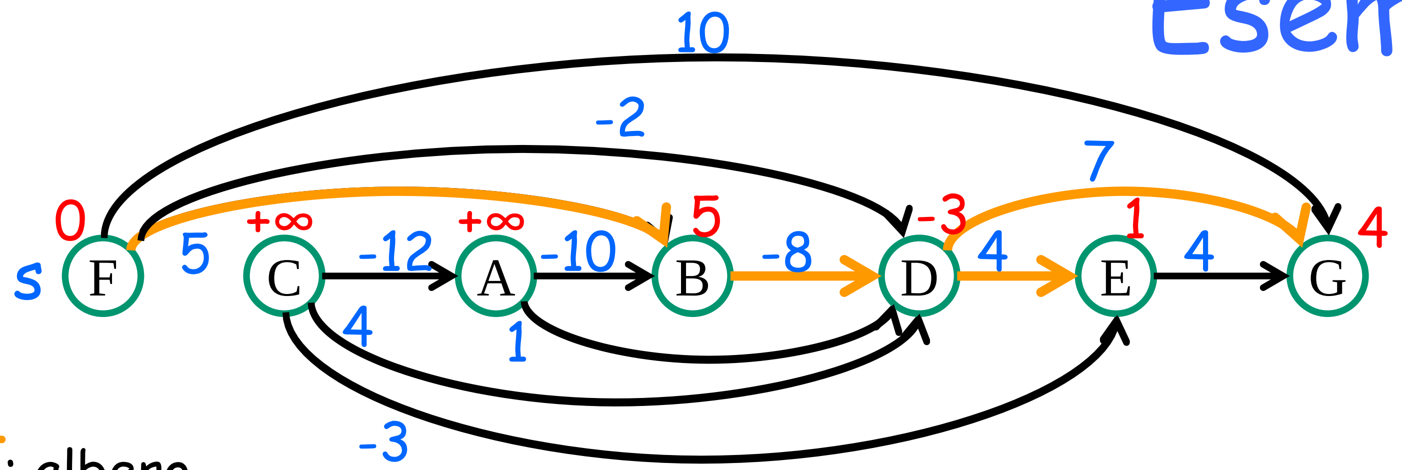
Tempo di esecuzione (liste di adiacenza): $\Theta(n+m)$

Esempio



stime distanze	F	C	A	B	D	E	G
inizio:	0	+∞	+∞	+∞	+∞	+∞	+∞
visito F	0	+∞	+∞	5	-2	+∞	10
visito C, A	0	+∞	+∞	5	-2	+∞	10
visito B	0	+∞	+∞	5	-3	+∞	10
visito D	0	+∞	+∞	5	-3	1	4
visito E	0	+∞	+∞	5	-3	1	4
visito G	0	+∞	+∞	5	-3	1	4

Esempio



T: albero dei cammini minimi radicato in s

	F	C	A	B	D	E	G
distanze da s	0	+∞	+∞	5	-3	1	4