Algoritmi e Strutture Dati

Capitolo 13

Cammini minimi:

Bellman e Ford

Punto della situazione

- Algoritmo basato sull'ordinamento topologico: albero dei cammini minimi (ACM) in grafi diretti aciclici. Complessità $\Theta(n+m)$ (con liste di adiacenza). Come si ottiene l'ordinamento topologico in $\Theta(n+m)$?
  - Basta associare inizialmente ad ogni nodo il proprio grado entrante. Tale valore può essere calcolato in $\Theta(n+m)$ scorrendo le liste di adiacenza. Quindi, manteniamo una lista Z dei nodi con grado entrante pari a 0. Tale lista si può costruire in $\Theta(n)$, ovviamente. Quindi, selezioniamo un elemento di Z, finché la lista non si svuota, e lo eliminiamo dal grafo; inoltre, per ogni arco uscente del nodo eliminato, cioè in tempo $\Theta(\delta(v))$ (il grado uscente di v) diminuiamo di 1 il grado entrante del nodo di arrivo corrispondente, aggiungendo eventualmente tale nodo a Z se il suo grado entrante si è azzerato.

Risoluzione secondo esercizio

Domanda: Quanto costa calcolare tutte le distanze da un nodo sorgente arbitrario in un albero non orientato e con pesi positivi di n nodi?

Risposta: $\Theta(n)$. Infatti, è sufficiente radicare l'albero in tale nodo e orientare tutti gli archi dalla radice verso le foglie, ottenendo ovviamente un grafo aciclico e fortemente connesso rispetto ai cammini uscenti dalla radice (cioè, la sorgente). Potremo quindi applicare l'algoritmo basato sull'ordinamento topologico, che costerà $\Theta(n+m)= \Theta(n)$ (poiché $m=n-1$). Si noti che più semplicemente, sfruttando il fatto che l'input è un albero, si può modificare la BFS o la DFS ottenendo lo stesso risultato (fatelo!).

Algoritmo di Bellman e Ford

(ACM in grafi diretti che non contengono cicli negativi o grafi non diretti che non contengono archi di costo negatico)

Richiamo: 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
- L'aggiornamento delle stime è basato sul seguente passo di rilassamento (π_{vy} denota un qualche cammino in G tra un generico nodo v e il nodo destinazione y; tale nodo v sarà selezionato secondo un qualche criterio indotto dall'algoritmo soggiacente):

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

Ordine di rilassamento

Supponiamo di aver inizializzato a $+\infty$ tutte le stime di distanza dei nodi di $G=(V,A,w)$ da s, escluso $D_{ss}=0$, e di dover trovare la distanza tra il nodo sorgente s ed un qualche nodo v di G. Sia $\pi_{sv}=<s,v_1,v_2,\dots,v_k = v>$ un cammino minimo in G tra s e v. Osserviamo che d_{sv} potrebbe ipoteticamente essere trovato eseguendo la seguente sequenza ottimale di k rilassamenti:

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

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

Approccio di Bellman e Ford

- Inizializza tutte le stime di distanza da s a $+\infty$, escluso $D_{ss}=0$, ed esegue n-1 passate
- In ciascuna passata, per ogni arco del grafo, esegui il relativo passo di rilassamento rispetto alla distanza dalla sorgente s (si noti ad esempio che 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 sicuramente 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 (ricordiamo che il grafo non contiene cicli negativi, e quindi per ogni coppia di nodi esiste sempre un cammini minimo semplice)

Pseudocodice

algoritmo `BellmanFord`($grafo$ *diretto* 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 ($(u,v) \in A$) **do**
      if $(D_{su} + w(u,v) < D_{sv})$ **then** $D_{sv} \leftarrow D_{su} + w(u,v)$
  return D

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

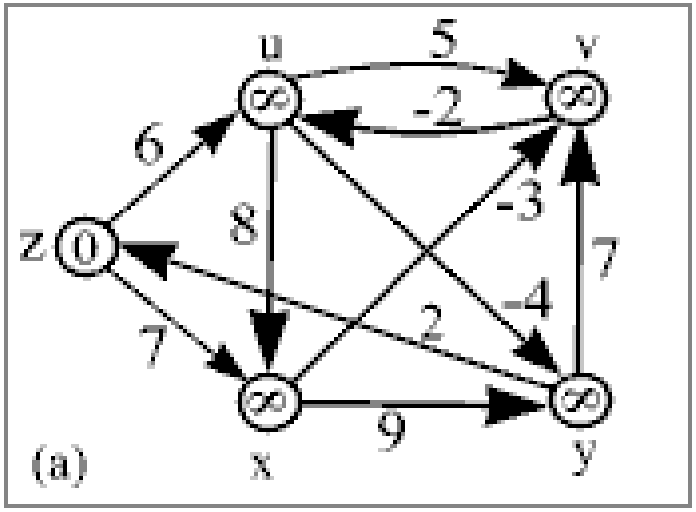
(con liste di adiacenza)

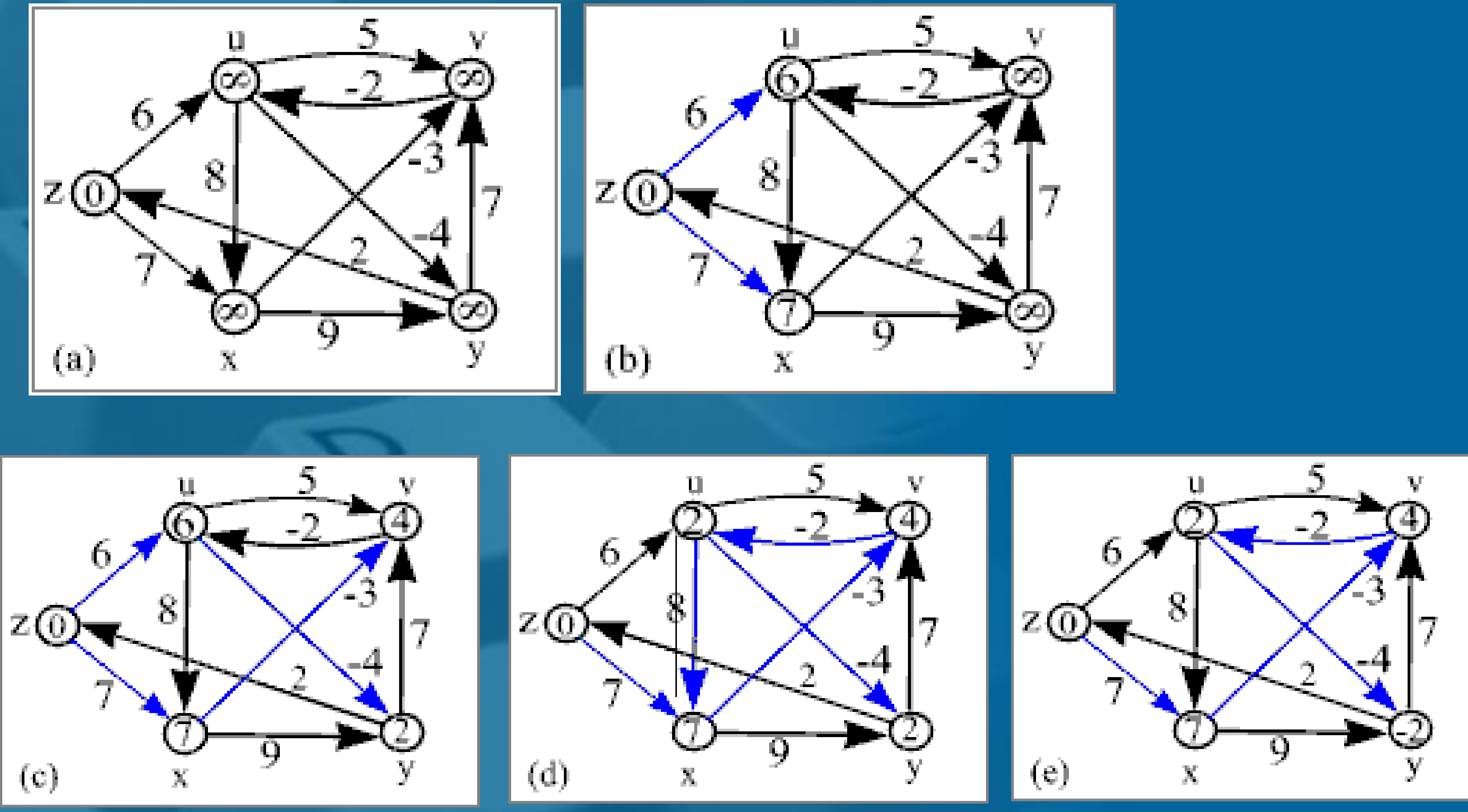
Quanto costerebbe se usassi una matrice di adiacenza per rappresentare il grafo? $\Theta(n^3)$

Per quali valori asintotici di m convengono le liste di adiacenza? $m=o(n^2)$

Esempio di esecuzione

Eseguire l'algoritmo di Bellman e Ford sul seguente grafo diretto, supponendo di partire dal nodo sorgente z, e ipotizzando di controllare gli archi del grafo in ordine lessicografico.




u v z x y
5 -2 6 8 -3 7 -4 2 7 9
(a) z 0, u ∞, v ∞, x ∞, y ∞
(b) z 0, u 6, v ∞, x 7, y ∞
(c) z 0, u 6, v 4, x 7, y 2
(d) z 0, u 2, v 4, x 7, y 2
(e) z 0, u 2, v 4, x 7, y -2