APA Modulo 1 Lezione 10

Elena Zucca

28 marzo 2020

Algoritmo di Floyd e Warshall

- problema: dato grafo orientato pesato trovare cammini minimi tra tutte le coppie di nodi
- sono ammessi costi negativi ma non cicli di costo negativo
- per semplicità identifichiamo i nodi con i numeri $1..n$
- idea: chiamiamo k-vincolato un cammino che passa solo per nodi in $1..k$ (esclusi gli estremi), per $k \leq n$
- $d^k(x, y) =$ distanza k-vincolata tra x e y, cioè la lunghezza minima di un cammino k-vincolato (come al solito, ∞ se tale cammino non esiste)
- è facile vedere che possiamo considerare solo i cammini semplici (senza nodi ripetuti)

Formulazione induttiva della soluzione: base

$$d^0(x,y) = \begin{cases} 0 & \textit{se } x = y \\ c_{x,y} & \textit{se } x \neq y \textit{ ed esiste l'arco } (x,y) \\ \infty & \textit{altrimenti} \end{cases}$$

Passo induttivo

$$d^k(x,y) = min\{d^{k-1}(x,y), d^{k-1}(x,k) + d^{k-1}(k,y)\}$$

infatti, dato un cammino minimo (semplice) k-vincolato da x a y, si hanno due casi:

- non passa per k, quindi è anche un cammino minimo $k-1$-vincolato
- passa per k, quindi (essendo semplice) è composto da un cammino $k-1$-vincolato da x a k, e da un cammino $k-1$-vincolato da k a y

Algoritmo di programmazione dinamica

utilizziamo $n+1$ matrici $n \times n$: $\mathtt{D}_0, \ldots, \mathtt{D}_n$

```
FloydWarshall(G)
  for each (x,y : nodi in G)
    D^0[x,y] =
        0 se x=y
        c_{x,y} se x ≠ y ed esiste arco (x,y)
        ∞ altrimenti
  for (k=1; k <= n; k++)
    for each (x,y : nodi in G)
      D^k[x,y] = D^{k-1}[x,y]
      if (D^{k-1}[x,k] + D^{k-1}[k,y]<D^k[x,y])
        D^k[x,y]= D^{k-1}[x,k] + D^{k-1}[k,y]
  return D^n
```

Ottimizzazione di spazio

- complessità temporale e spaziale cubica
- è possibile utilizzare uno spazio quadratico anziché cubico ossia, utilizzare un'unica matrice `D`
- problema: in $\mathtt{D}^k\mathtt{[x,y]} = \mathtt{D}^{k-1}\mathtt{[x,k]} + \mathtt{D}^{k-1}\mathtt{[k,y]}$, con unica matrice questi valori potrebbero essere già stati aggiornati, cioè essere $\mathtt{D}^k\mathtt{[x,k]}$ e $\mathtt{D}^k\mathtt{[k,y]}$
- ma questo è ininfluente in quanto questi valori aggiornati sono necessariamente uguali ai precedenti

Infatti:

- stiamo calcolando la distanza vincolata da un nodo x a k (e da k a un nodo y)
- quindi aggiungere il nodo k tra quelli utilizzabili non cambia le cose formalmente:

  $$d^k(x,k) = min\{d^{k-1}(x,k), d^{k-1}(x,k) + d^{k-1}(k,k)\} = d^{k-1}(x,k)$$

- analogamente $d^k(k,y)$

Quindi:

```
FloydWarshall(G)
  for each (x,y : nodi in G)
    D[x,y] =
        0 se x=y,
        c_{x,y} se x ≠ y ed esiste arco (x,y)
        ∞ altrimenti
  for each (k=1; k <= n; k++)
    for (x,y : nodi in G)
      if (D[x,k] + D[k,y] < D[x,y])
        D[x,y] = D[x,k] + D[k,y]
  return D
```

Matrice dei predecessori

se vogliamo ottenere, oltre alla distanza, anche il cammino minimo, definiamo

> $\pi_{xy} =$ *null se* $x = y$ *oppure non esiste un cammino da* x *a* y*,*
> *altrimenti è il predecessore di* y *in un cammino minimo da* x *a* y

Formulazione induttiva del predecessore: base

$$\pi^0(x, y) = \begin{cases} x & se\ x \neq y\ ed\ esiste\ l'arco\ (x, y) \\ null & altrimenti \end{cases}$$

Passo induttivo

$$\pi^k(x, y) = \begin{cases} \pi^k(k, y) & \textit{se } D[x, k] + D[k, y] < D[x, y] \\ \pi^{k-1}(x, y) & \textit{altrimenti} \end{cases}$$

Algoritmo con matrice dei predecessori

```
FloydWarshall(G)
  for each (x,y : nodi in G)
    D[x,y] =
        0 se x=y,
        c_{x,y} se x ≠ y ed esiste arco (x,y)
        ∞ altrimenti
    Π[x,y] =
        x se x ≠ y ed esiste arco (x,y)
        null  altrimenti
  for (k=1; k <= n; k++)
    for each (x,y : nodi in G)
      if (D[x,k] + D[k,y] < D[x,y])
        D[x,y] = D[x,k] + D[k,y]
        Π[x,y] = Π[k,y]
return D,Π
```

Sottografo dei cammini minimi

- la matrice dei predecessori finale individua il sottografo dei cammini minimi:
    - i nodi sono tutti i nodi
    - gli archi sono tutti quelli da un predecessore a un nodo raggiungibile ossia della forma (π_{xy}, y) con $\pi_{xy} \neq \texttt{null}$

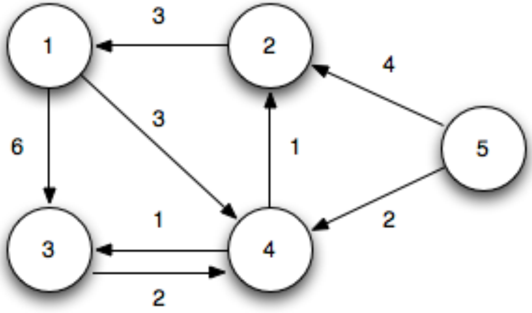
Albero dei cammini minimi a partire da un nodo

- il sottografo indotto dalla riga x della matrice Π è l'albero dei cammini minimi con radice x
  - i nodi sono tutti i nodi raggiungibili da x
    ossia gli y tali che $\pi_{xy} \neq \texttt{null}$, più x stesso
  - gli archi sono tutti quelli da un predecessore a un nodo raggiungibile
    ossia della forma (π_{xy}, y) con $\pi_{xy} \neq \texttt{null}$

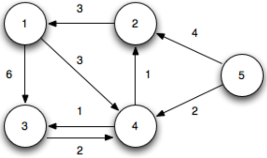
Cammino minimo da x a y

```
shortest_path(Π,x,y)
  if (x=y) return x
  else if (Π[x,y] = null)
    return ... [non esiste cammino]
  else return shortest_path(Π,x,Π[x,y]) · y
```

Esempio



Passo 0: non uso nodi intermedi



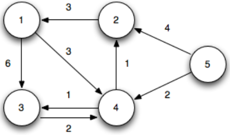
D

	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	∞	∞	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

Π

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	/	/	/
3	/	/	/	3	/
4	/	4	4	/	/
5	/	5	/	5	/

Passo 1: uso il nodo 1

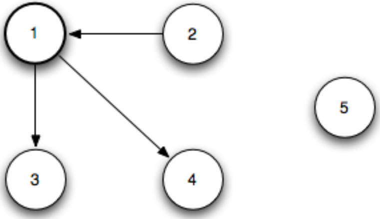


D

	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	∞	∞	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

Π

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	/	/	/
3	/	/	/	3	/
4	/	4	4	/	/
5	/	5	/	5	/



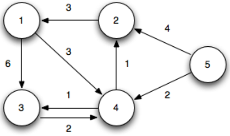
D

	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	**9**	**6**	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

Π

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	**1**	**1**	/
3	/	/	/	3	/
4	/	4	4	/	/
5	/	5	/	5	/

Passo 2: uso (anche) il nodo 2

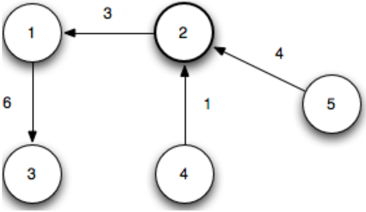


D

	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	9	6	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

Π

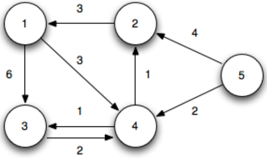
	1	2	3	4	5
1	/	/	1	1	/
2	2	/	1	1	/
3	/	/	/	3	/
4	/	4	4	/	/
5	/	5	/	5	/



	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	9	6	∞
3	∞	∞	0	2	∞
4	**4**	1	1	0	∞
5	**7**	4	**13**	2	0

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	1	1	/
3	/	/	/	3	/
4	**2**	4	4	/	/
5	**2**	5	**1**	5	/

Passo 3: uso (anche) il nodo 3



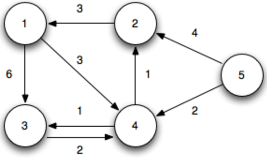
	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	9	6	∞
3	∞	∞	0	2	∞
4	**4**	1	1	0	∞
5	**7**	4	**13**	2	0

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	1	1	/
3	/	/	/	3	/
4	**2**	4	4	/	/
5	**2**	5	**1**	5	/

	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	9	6	∞
3	∞	∞	0	2	∞
4	**4**	1	1	0	∞
5	**7**	4	**13**	2	0

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	1	1	/
3	/	/	/	3	/
4	**2**	4	4	/	/
5	**2**	5	**1**	5	/

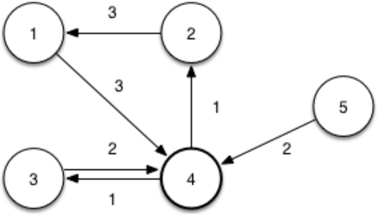
Passo 4: uso (anche) il nodo 4



	1	2	3	4	5
1	0	∞	6	3	∞
2	3	0	9	6	∞
3	∞	∞	0	2	∞
4	**4**	1	1	0	∞
5	**7**	4	**13**	2	0

	1	2	3	4	5
1	/	/	1	1	/
2	2	/	1	1	/
3	/	/	/	3	/
4	**2**	4	4	/	/
5	**2**	5	**1**	5	/

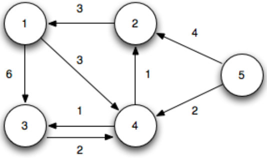
D Π



	1	2	3	4	5
1	0	**4**	**4**	3	∞
2	3	0	**7**	6	∞
3	**6**	**3**	0	2	∞
4	4	1	1	0	∞
5	**6**	**3**	**3**	2	0

	1	2	3	4	5
1	/	**4**	**4**	1	/
2	2	/	**4**	1	/
3	**2**	**4**	/	3	/
4	2	4	4	/	/
5	2	**4**	**4**	5	/

Passo 4: uso (anche) il nodo 5



D

	1	2	3	4	5
1	0	**4**	**4**	3	∞
2	3	0	**7**	6	∞
3	**6**	**3**	0	2	∞
4	4	1	1	0	∞
5	**6**	**3**	**3**	2	0

Π

	1	2	3	4	5
1	/	**4**	**4**	1	/
2	2	/	**4**	1	/
3	**2**	**4**	/	3	/
4	2	4	4	/	/
5	2	**4**	**4**	5	/

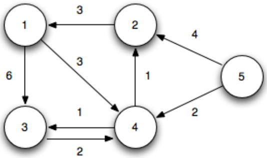
D

	1	2	3	4	5
1	0	**4**	**4**	3	∞
2	3	0	**7**	6	∞
3	**6**	**3**	0	2	∞
4	4	1	1	0	∞
5	**6**	**3**	**3**	2	0

Π

	1	2	3	4	5
1	/	**4**	**4**	1	/
2	2	/	**4**	1	/
3	**2**	**4**	/	3	/
4	2	4	4	/	/
5	2	**4**	**4**	5	/

Risultato finale



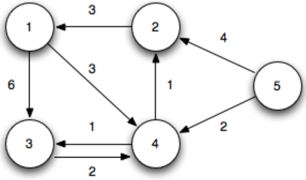
D

	1	2	3	4	5
1	0	**4**	**4**	3	∞
2	3	0	**7**	6	∞
3	**6**	**3**	0	2	∞
4	4	1	1	0	∞
5	**6**	**3**	**3**	2	0

Π

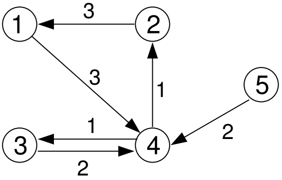
	1	2	3	4	5
1	/	**4**	**4**	1	/
2	2	/	**4**	1	/
3	**2**	4	/	3	/
4	2	4	4	/	/
5	2	**4**	**4**	5	/

Sottografo dei cammini minimi



Π

	1	2	3	4	5
1	/	4	4	1	/
2	2	/	4	1	/
3	2	4	/	3	/
4	2	4	4	/	/
5	2	4	4	5	/



Albero dei cammini minimi a partire dal nodo 1