ALGORITMO DI FLOYD-WARSHALL

ALGORITMI E STRUTTURE DATI

ALGORITMO DI FLOYD-WARSHALL

- L'algoritmo di Floyd-Warshall è un algoritmo per trovare il percorso più corto tra ogni coppia di nodi in un grafo
- Ovvero, dato un grafo $G = (V, E)$ avremo come risultato una tabella D in cui in posizione (i, j) è la lunghezza del percorso di peso/costo minimo tra i e j.
- Il grafo può essere orientato, pesato e anche con pesi negativi sugli archi

ALGORITMO DI FLOYD-WARSHALL

- Supponiamo che i nodi corrispondano agli interi $\{0,\ldots,k\}$
- Proviamo a definire in modo ricorsivo il percorso tra due nodi i e j
- Per fare questo consideriamo una restrizione sui nodi intermedi che possono essere contenuti nel percorso tra i e j
- Indichiamo con $d_{i,j}^{k}$ la lunghezza del percorso più corto tra i e j che contenga nodi intermedi solo nodi in $\{0,\ldots,k-1\}$

ALGORITMO DI FLOYD-WARSHALL

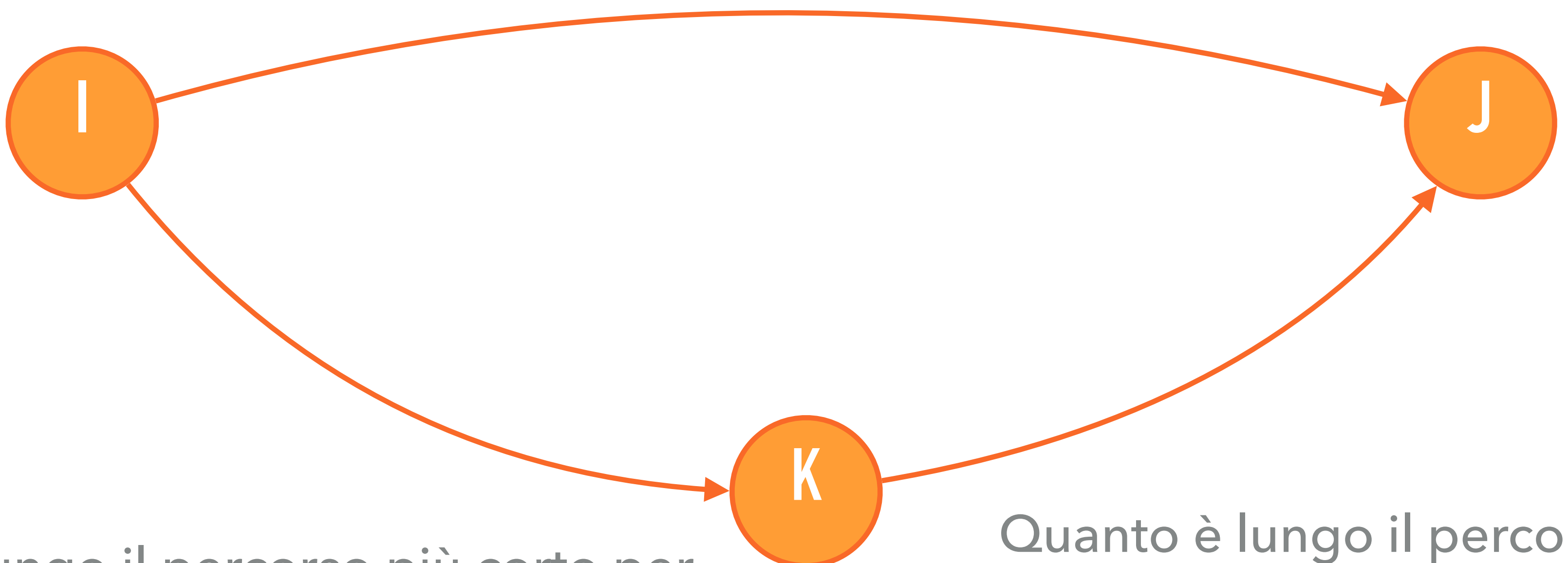
- Il caso base è quando non possiamo avere nodi intermedi, quindi $d^0_{i,j}$.
- Dato che non possiamo avere nodi intermedi ci sono solo tre possibilità:
  - $d^0_{i,i} = 0$, dato che nodo di destinazione e partenza sono lo stesso
  - $d^0_{i,j} = w_{i,j}$ se esiste un arco tra i e j
  - $d^0_{i,j} = +\infty$ se nessun arco connette i con j

ALGORITMO DI FLOYD-WARSHALL

- Per un generico $d_{i,j}^{k+1}$ abbiamo due casi:
  - Il nodo k non viene usato nel percorso di lunghezza minima per andare da i a j: in quel caso $d_{i,j}^{k+1} = d_{i,j}^{k}$
  - Il nodo k viene usato nel percorso di lunghezza minima per andare da i a j. Dato che un nodo può apparire una sola volta nel percorso non useremo k per andare da i a k e per andare da k a j, quindi $d_{i,j}^{k+1} = d_{i,k}^{k} + d_{k,j}^{k}$

ALGORITMO DI FLOYD-WARSHALL

Quanto è lungo il percorso più corto per andare da i a j usando solo $\{0,\ldots k\}$ come nodi intermedi e senza passare per k? $d^k_{i,j}$



Quanto è lungo il percorso più corto per andare da i a k usando solo $\{0,\ldots k\}$ come nodi intermedi? $d^k_{i,k}$

Quanto è lungo il percorso più corto per andare da k a j usando solo $\{0,\ldots k\}$ come nodi intermedi? $d^k_{k,j}$

Totale: $d^k_{i,k} + d^k_{k,j}$

ALGORITMO DI FLOYD-WARSHALL

- Abbiamo quindi che $d_{i,j}^{k+1} = \min\{d_{i,j}^{k}, d_{i,k}^{k} + d_{k,j}^{k}\}$ dato che abbiamo solo due scelte possibili e vogliamo il percorso di lunghezza/peso minimo
- Possiamo iniziare da $k = 0$, in cui conosciamo tutti i valori, fino ad arrivare a $k = n$, in cui possiamo utilizzare tutti gli n vertici del grafo come nodi intermedi

PSEUDOCODICE: FLOYD-WARSHALL

Parametri: Matrice di adiacenza del grafo W di n nodi con $+\infty$ dove gli archi sono assenti

D = array di dimensione $(n+1) \times n \times n$ # contiene tutti i $d_{i,j}^{k}$

D[0] = W # la matrice W contiene già i casi base

for k in range(0,n):

    # calcoliamo $d_{i,j}^{k+1}$ al variare di i e j

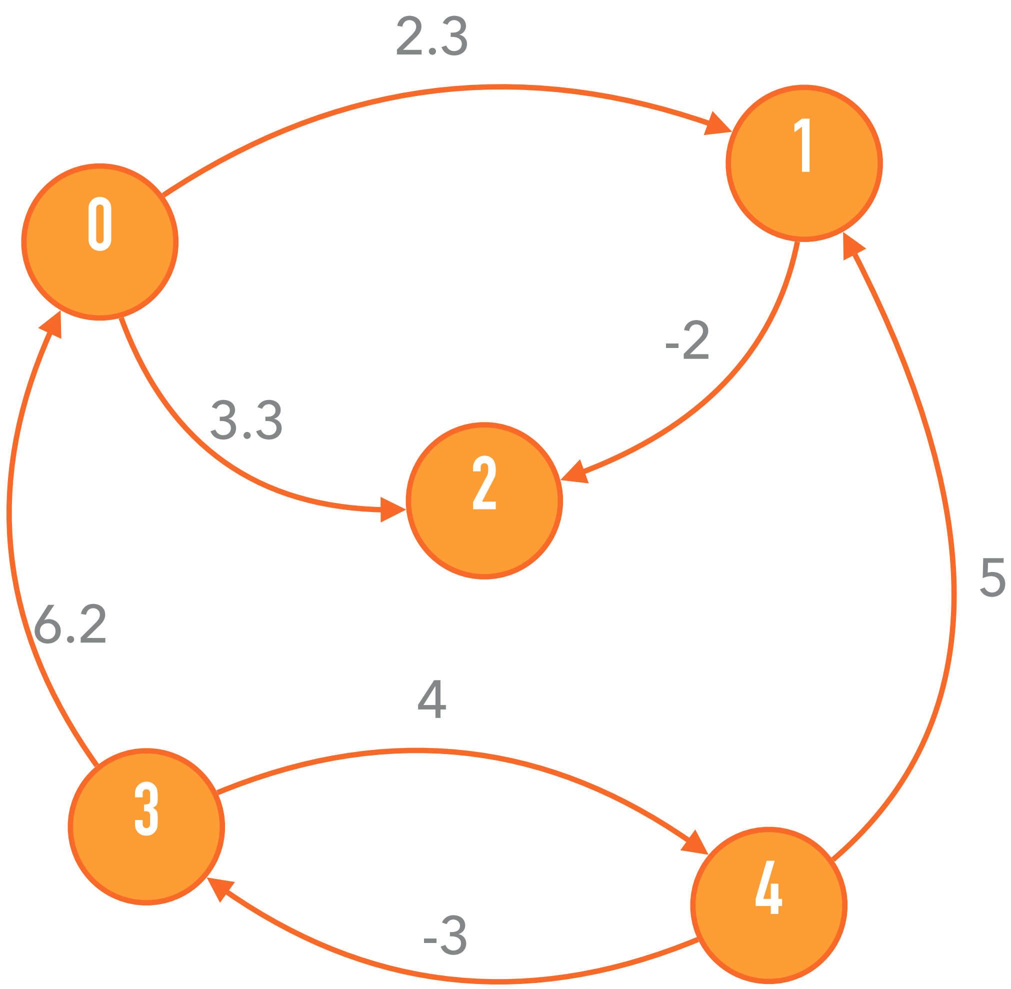
    for i in range(0,n):

        for j in range(0,n):

            D[k+1][i][j] = min(D[k][i][j], D[k][i][k] + D[k][k][j])

return D[n]

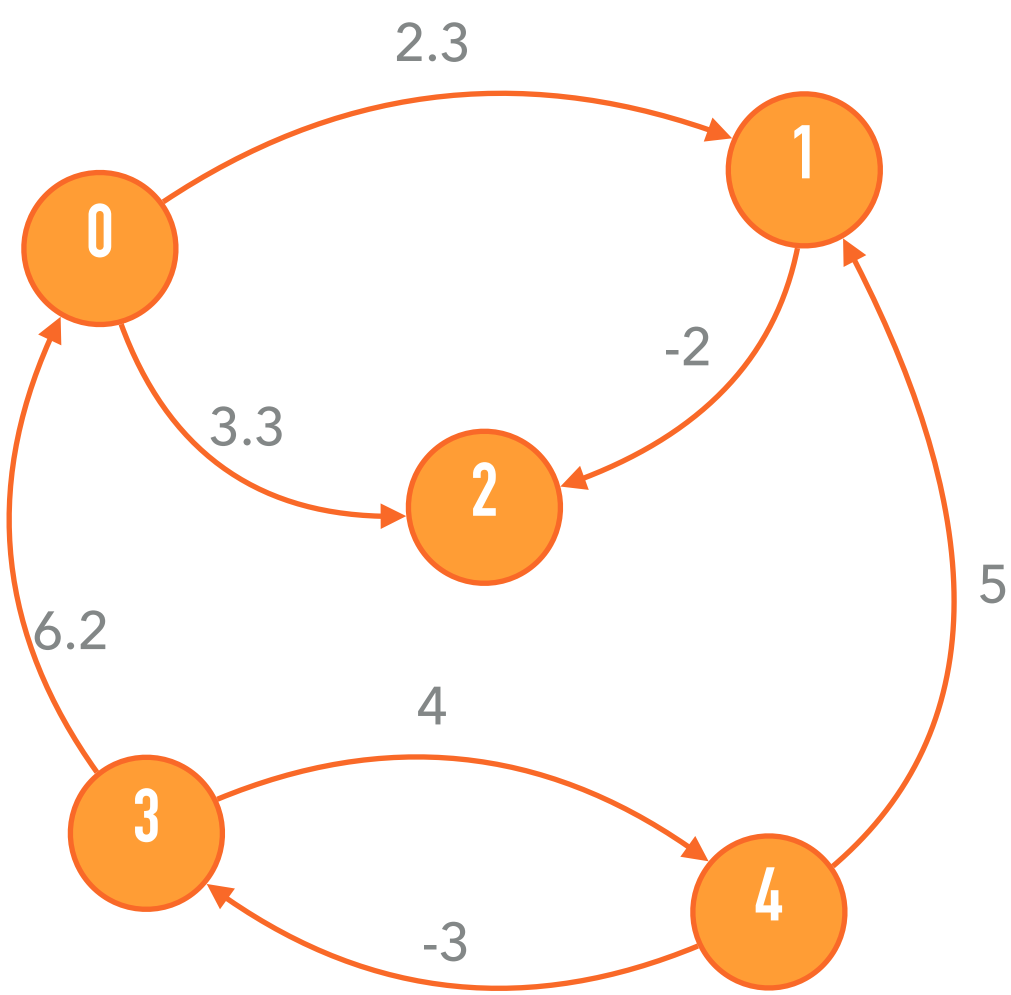
ALGORITMO DI FLOYD-WARSHALL



W=D[0]=

0	2.3	3.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	∞	∞	0	4
∞	5	∞	-3	0

ALGORITMO DI FLOYD-WARSHALL



W=D[0]=

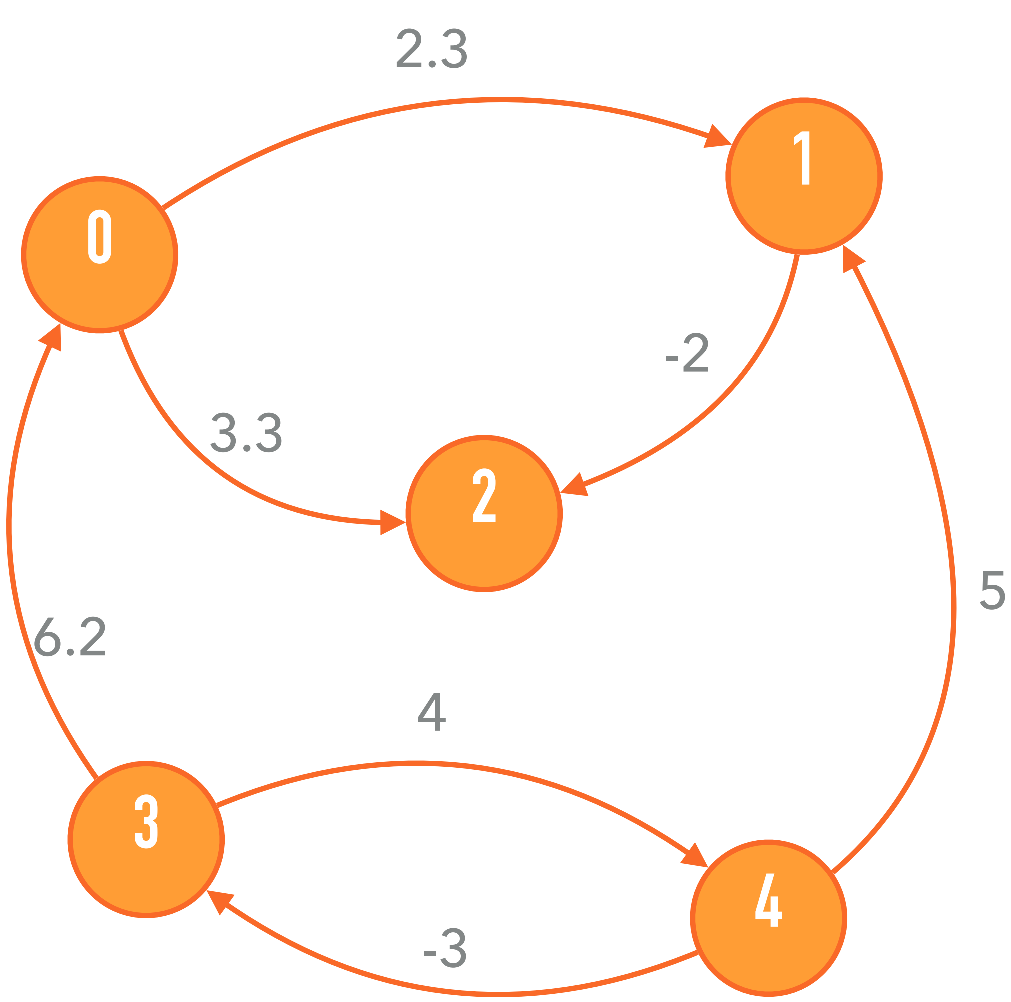
0	2.3	3.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	∞	∞	0	4
∞	5	∞	-3	0

D[1]=

0	2.3	3.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	9.5	0	4
∞	5	∞	-3	0

Possiamo passare per il nodo 0

ALGORITMO DI FLOYD-WARSHALL



D[1]=

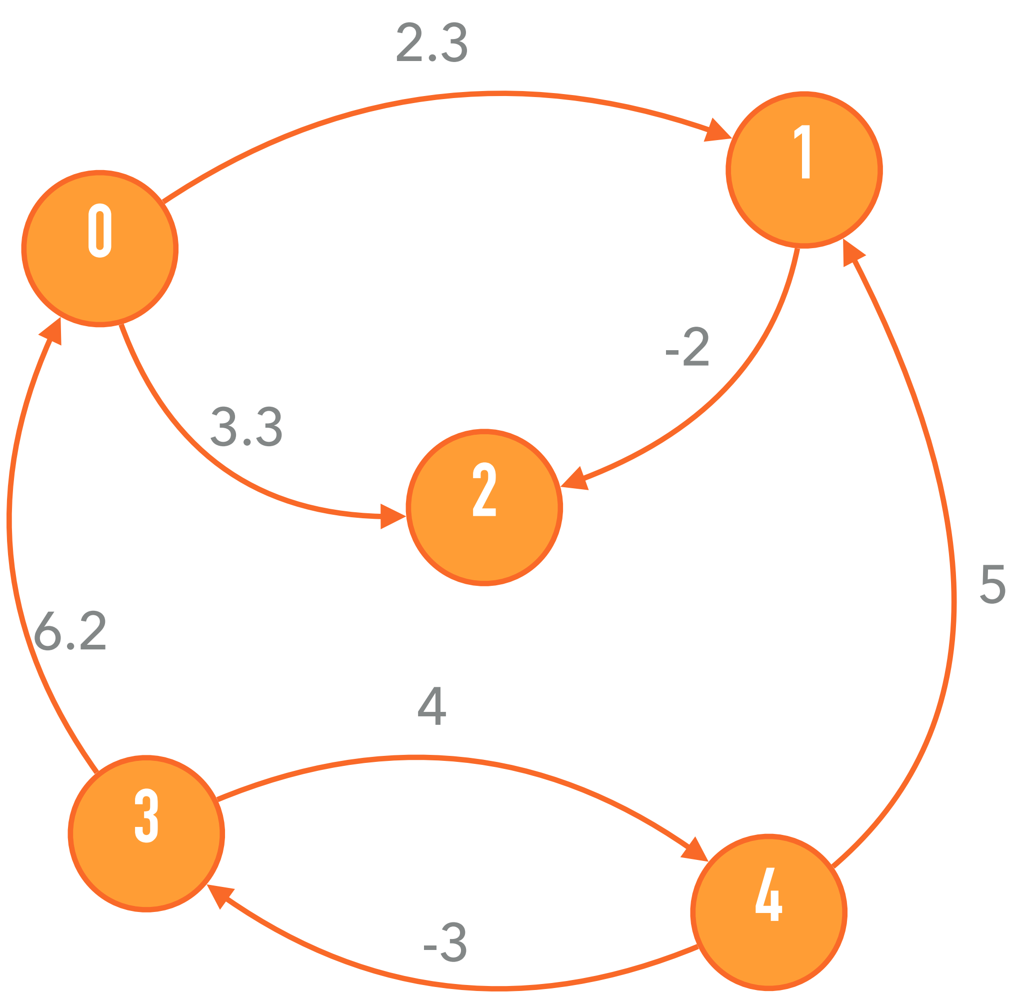
0	2.3	3.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	9.5	0	4
∞	5	∞	-3	0

D[2]=

0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
∞	5	3	-3	0

Possiamo passare per i nodi 0 e 1

ALGORITMO DI FLOYD-WARSHALL



D[2]=

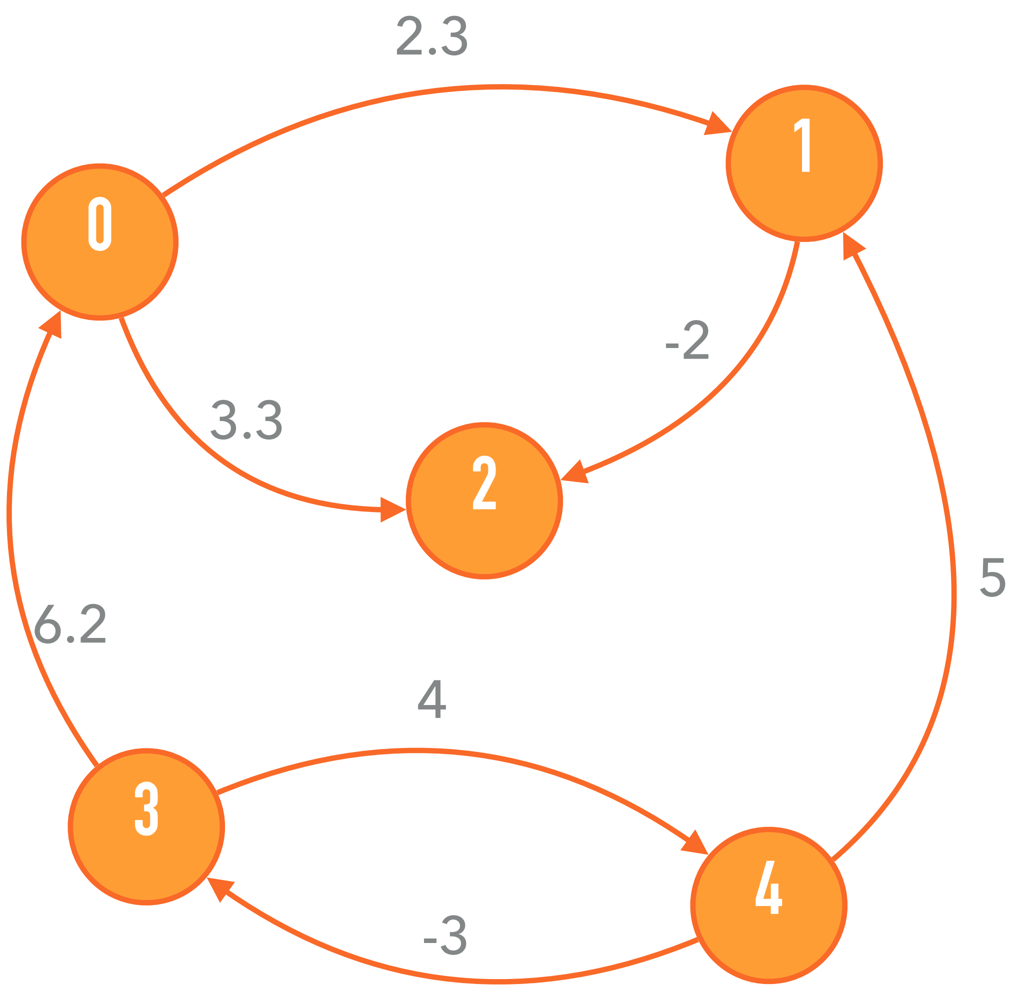
0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
∞	5	3	-3	0

D[3]=

0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
∞	5	3	-3	0

Possiamo passare per i nodi 0, 1 e 2

ALGORITMO DI FLOYD-WARSHALL



D[3]=

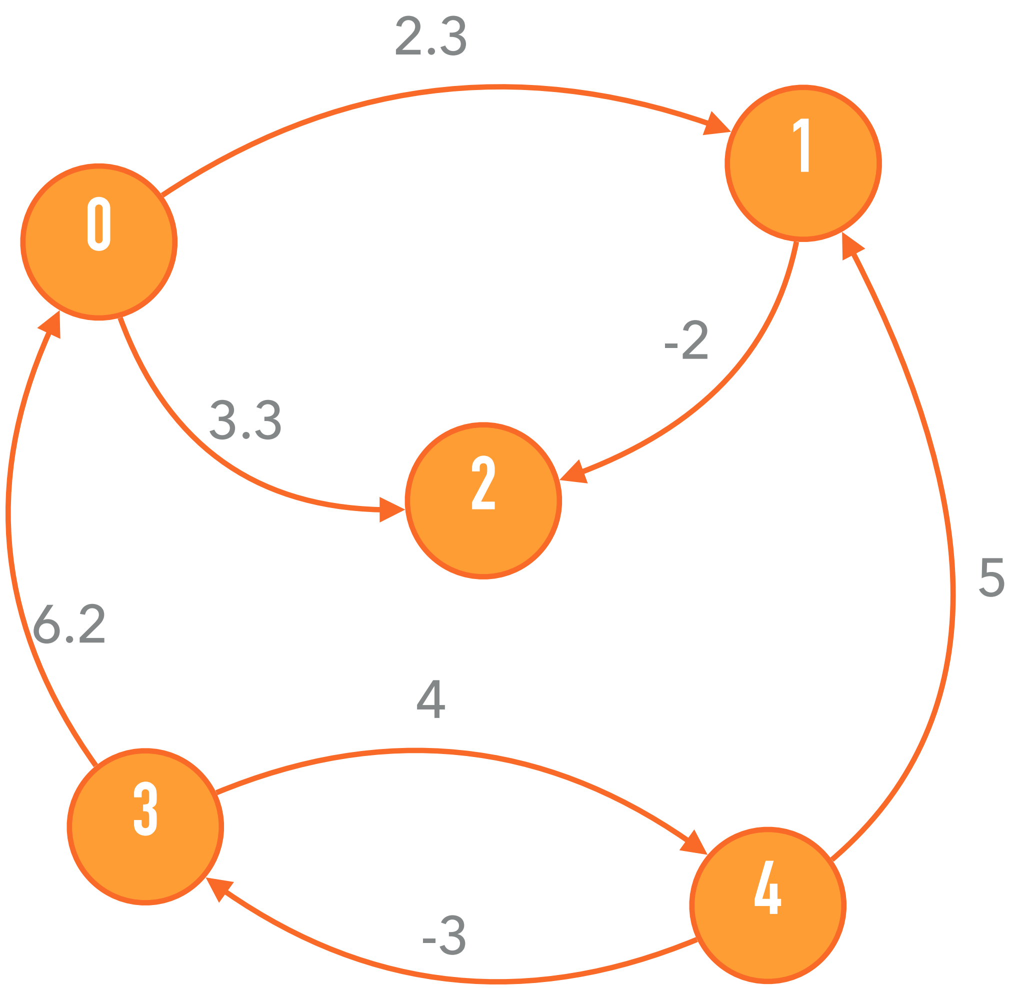
0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
∞	5	3	-3	0

D[4]=

0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
3.2	5	3	-3	0

Possiamo passare per i nodi 0, 1, 2 e 3

ALGORITMO DI FLOYD-WARSHALL



D[4]=

0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
3.2	5	3	-3	0

D[5]=

0	2.3	0.3	∞	∞
∞	0	-2	∞	∞
∞	∞	0	∞	∞
6.2	8.5	6.5	0	4
3.2	5	3	-3	0

Possiamo passare per i nodi 0, 1, 2, 3 e 4

Matrice con la lunghezza dei percorsi di costo minimo tra tutte le coppie di vertici

ALGORITMO DI FLOYD-WARSHALL: COMPLESSITÀ

- Abbiamo tre cicli for innestati e ognuno di questi cicli for svolge $|V| = n$ iterazioni
- Ne segue che il tempo di esecuzione è $O(V^3)$, quindi cubico
- Notiamo che rispetto all'algoritmo di Bellman-Ford, che richiede tempo $O(VE)$ con l'algoritmo di Floyd-Warshall otteniamo la distanza tra ogni coppia di vertici, non solo a partire da un nodo sorgente dato

ALGORITMO DI FLOYD-WARSHALL: VARIANTI

- Possiamo aggiungere molteplici restrizioni e varianti all'algoritmo di Floyd-Warshall a seconda delle necessità
- Come esempio, pensiamo di voler stabilire per ogni coppia di nodi i e j il cammino semplice meno costoso per andare da i a j che però contenga un numero pari di nodi
- Come possiamo modificare l'algoritmo?

ALGORITMO DI FLOYD-WARSHALL: VARIANTI

- Teniamo traccia di una informazione aggiuntiva: se il cammino che abbiamo è con un numero pari o dispari di nodi:
  - $d_{i,j}^{k,\text{pari}}$ è il costo minimo per andare da i a j passando per un numero pari di nodi usando solo $\{0,\ldots,k-1\}$ come nodi intermedi
  - $d_{i,j}^{k,\text{dispari}}$ è definito in modo simile ma con un numero dispari di nodi

ALGORITMO DI FLOYD-WARSHALL: VARIANTI

- I casi base sono:
  - $d_{i,i}^{0,\text{pari}} = 0$
  - $d_{i,j}^{0,\text{pari}} = w_{i,j}$ se esiste l'arco (i, j)
  - $d_{i,j}^{0,\text{pari}} = + \infty$ altrimenti
  - $d_{i,j}^{0,\text{dispari}} = + \infty$ dato che senza nodi intermedi ogni percorso ha solo un numero pari di nodi intermedi

ALGORITMO DI FLOYD-WARSHALL: VARIANTI

- Il calcolo delle soluzioni di sotto-problemi cambia solo per mantenere corretta la parità:
- $d_{i,j}^{k+1,\text{pari}} = \min\{d_{i,j}^{k,\text{pari}}, d_{i,k}^{k,\text{dispari}} + d_{k,j}^{k,\text{pari}}, d_{i,k}^{k,\text{pari}} + d_{k,j}^{k,\text{dispari}}\}$
- $d_{i,j}^{k+1,\text{dispari}} = \min\{d_{i,j}^{k,\text{dispari}}, d_{i,k}^{k,\text{pari}} + d_{k,j}^{k,\text{pari}}, d_{i,k}^{k,\text{dispari}} + d_{k,j}^{k,\text{dispari}}\}$
- Ovvero trattiamo in modo esplicito come combinare percorsi di parità diversa cambia la parità del percorso finale

ALGORITMO DI FLOYD-WARSHALL: ESERCIZI

- Alcune varianti che si possono provare a definire, come esercizio:
- Dato un grafo stabilire il costo del percorso di lunghezza minima che contiene un numero **dispari** di **archi**.
- In questo caso cambia un poco la ricorrenza (e i casi base) perché contiamo gli archi e non i nodi