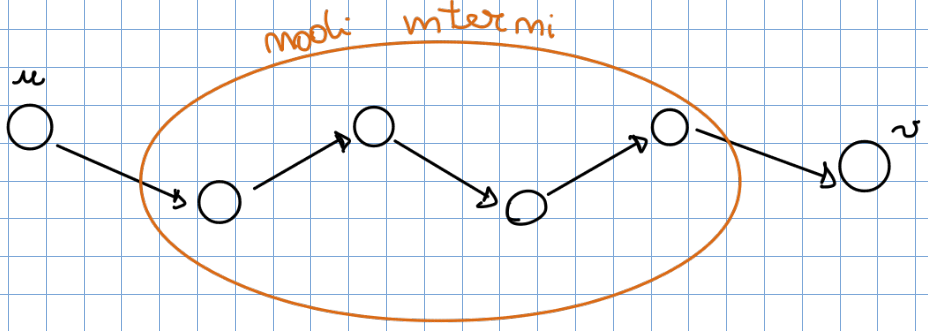
Simulazione: aula 24 dalle 9-12 giorno 21

Floyd-Warshall

d = # massimo di archi in un c.m



$V = \{ v_1, v_2, v_3 \dots v_n \}$

$V_k = \{ v_1, v_2, v_3 \dots v_k \}$

$V_0 = \{ \}$

$V_1 = \{ v_1 \}$

$V_2 = \{ v_1, v_2 \}$

$\vdots$

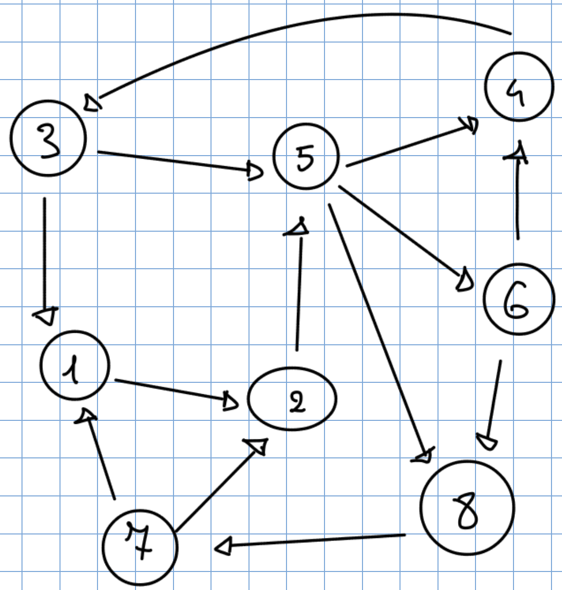
$V_n = V$

notazione per indicare il sottoinsieme fatto da k nodi

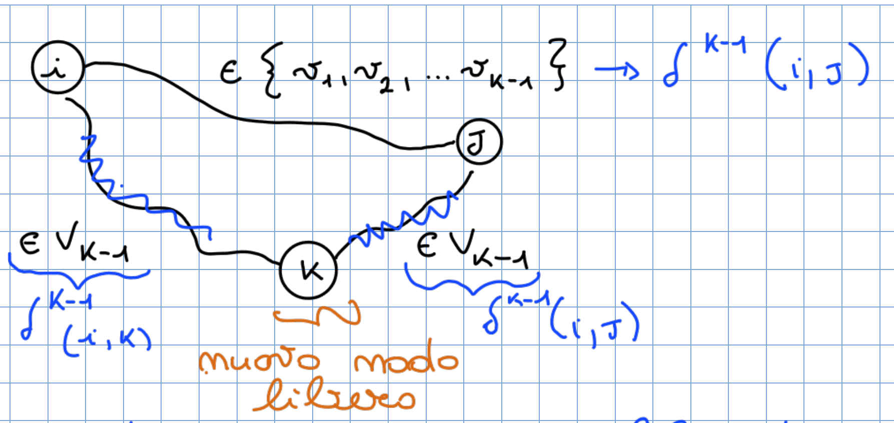
la dimensione del problema diventa: il numero di cammini minimi con k nodi

$\delta^0(i,j)$ non faccio niente

$\delta^1(i,j)$ posso passare da 1



accompagniamo la definizione di dimensione del problema con:



al K-esimo passaggio avrò solo un nodo K che sarà l'ultimo aggiunto a V_K

unendo i due cammini blu ottengo un cammino che si basa su K

$$\underbrace{D^K[i,j]}_{\text{una matrice per mantenere i valori ad ogni passo}} = \delta^K(i,j)$$

$$D^k[i,j] = \begin{cases} w[i,j] & \text{se } K = 0 \quad \text{caso base con } V^0 = \{\} \\ \underbrace{\min}_{\text{non ho bisogno di un for che scorre tutti i nodi perché } k \text{ è definito}}\left(D^{K-1}[i,j],\ D^{K-1}[i,k] + D^{K-1}[k,j]\right) \end{cases}$$

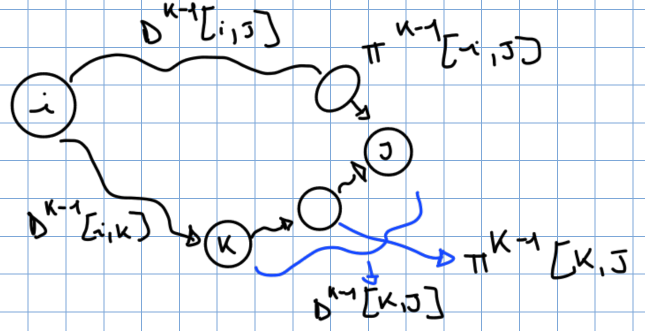
matrice di adiacenze

```
Floyd-Warshall(W)
   D^0 = W
   for K ← 1 to n Do                          O(n)
      D^K ← new Matrix(n, n)
      for i ← 1 to n Do                       O(n)
         for j ← 1 to n do                    O(n)
            D^K[i,j] ← D^{K-1}[i,j]
            if D^K[i,j] > D^{K-1}[i,K] + D^{K-1}[K,j]
            then D^K[i,j] ← D^{K-1}[i,K] + D^{K-1}[K,j]
   return D^n
```

$O(V^3)$

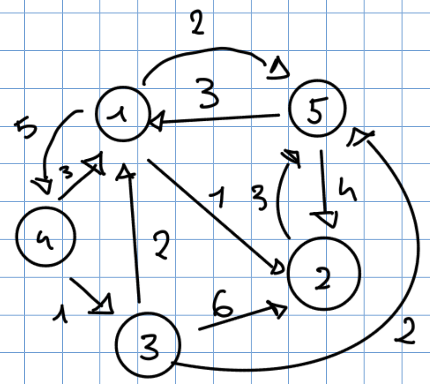
Se mantieniamo $\pi[i,j]$ ovvero il cammino minimo predecessore ottieniamo un albero dei cammini minimi

$$\pi^0[i,j] \begin{cases} i & \text{se } [i,j] \in E \\ \text{null} & \text{altrimenti} \end{cases}$$




```
Floyd-Warshall (W)
    D^0 = W
    π[i,j] = { i   se [i,j] ∈ E
             { null
    for k <- 1 to n Do
        D^k <- new Matrix(n,n)
        for i <- 1 to n Do
          for j <- 1 to n do
              D^k[i,j] <- D^(k-1)[i,j]
              π^k[i,j] <- π^(k-1)[i,j]
              if D^k[i,j] > D^(k-1)[i,k] + D^(k-1)[k,j]
              then D^k[i,j] <- D^(k-1)[i,k] + D^(k-1)[k,j]
                   π^k[i,j] <- π^(k-1)[k,j]
    return D^n
```

Esempio



$W = D^0$

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

stesse cose per le col

↳ prendo queste col, se c'è inf in una riga le riscrivo uguale
se c'è 0 riscrivo
tutti quelli in arrivo su K

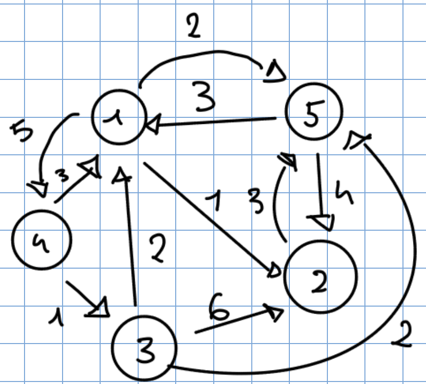
questo è K

$W = D^1$

	1	2	3	4	5
1	0	1	∞	5	2
2	∞	0	∞	∞	3
3	2	3	0	7	2
4	3	4	1	0	5
5	3	4	∞	8	0

$W = D^3$

	1	2	3	4	5
1	0	1	∞	5	2
2	∞	0	∞	∞	3
3	2	3	0	7	2
4	3	4	1	0	3
5	3	4	∞	8	0



$W = D^2$

	1	2	3	4	5
1	0	1	∞	5	2
2	∞	0	∞	∞	3
3	2	3	0	7	2
4	3	4	1	0	5
5	3	4	∞	8	0

$W = D^4$

	1	2	3	4	5
1	0	1	6	5	2
2	∞	0	∞	∞	3
3	2	3	0	7	2
4	3	4	1	0	3
5	3	4	9	8	0

$W = D^5$

	1	2	3	4	5
1	0	1	6	5	2
2	6	0	12	11	3
3	2	3	0	7	2
4	3	4	1	0	3
5	3	4	5	8	0

$$D_5[i,j] = \delta(i,j)$$