All pair shortest path

Vogliamo trovare i cammini minimi tra tutte le coppie ed ottenere l'output in forma tabulare
Invece di applicare |V| volte un algoritmo per i cammini minimi, possiamo fare di meglio con la programmazione dinamica

Useremo la variabile m → indica il numero massimo di archi ammessi in un cammino

se $m = 0$ significa che sono già nella destinazione $i = j$ altrimenti ∞

se $m = 1$ ho la matrice di adiacenza W che sarà D^1

se $m > 1$: $D^m[i,j] = \min\limits_{0 \le k < m} \big(\underbrace{D^{m-1}[i,k]}_{\text{cammino minimo fino a } m-1} + \underbrace{W[k,j]}_{\text{matrice adiacente } D^1} \big)$

EXTEND-SHORTEST-PATH
Date le matrici D^{m-1} e W restituisce D^m quindi aggiunge un arco per volta (crea quindi matrici D^2, D^3 fino ad arrivare ad $m-1$)

```
EXTEND-SHORTEST-PATHS (D^{m-1}, W)
   n = D.rows
   D^m = new matrix (n x n)
   FOR i <- 1 TO n
      FOR J <- 1 TO n
         D^m[i,J] = ∞
         FOR K <- 1 TO n
            IF ( D^m[i,J] > D^{m-1}[i,K] + W[K,J] )
```

```
        if ( D[i,j] > D[i,k] + W[k,j] )
            D^m[i,j] ← D^(m-1)[i,k] + W[k,j]
RETURN D^m
```

Alla fine della procedura avremo aggiunto un arco

Tempo $O(n^3)$ per i cicli for annidati

```
SLOW-APSP (W)
  n ← W.rows
  D^1 ← W
  FOR m ← 2 TO n-1 DO
      D^m ← EXTENDED-APSP(D^(m-1), W)
  RETURN D
```

$O(n^4)$

Posso migliorare sfruttando la sottostruttura ottima ed estendendo del doppio la matrice

$$
\begin{aligned}
D^1 &= W \\
D^2 &= D^1 \times D^1 \\
D^4 &= D^2 \times D^2 \\
D^8 &= D^4 \times D^4
\end{aligned}
$$

a partire da $n-1$ le matrici saranno tutte uguali

tecnica dell'elevazione al quadrato ripetuta

```
FAST-APSP (W)
  n ← W.rows
  D^1 ← W
  m ← 1
  while ( m < n-1 )
```

$D^{2m} = \text{EXTENDED-APSP}(D^m, D^m)$

$m = 2m$

RETURN D^m

$O(n^3 \log n)$