CHEMINS DE POIDS MINIMAL ENTRE TOUS COUPLES DE SOMMETS

But

- Pour chaque couple de sommet (x,y), on souhaite connaitre le chemin de poids minimal d'origine x et d'extrémité y dans le graphe.

- Utile pour l'acheminement d'information d'une source vers une destination dans un réseau (table de routage)

Par opérations sur les matrices

- On calcule pour chaque couple de sommet (x,y) la valeur

$$v2(xy) = \mathrm{Min}_{z \in X}(v(xz)+v(zy))$$

- On obtient ainsi le chemin de poids minimal de x à y dont la longueur est au plus 2. Si on souhaite les chemins de longueur au plus 3 on calcule la valeur :

$$v3(xy) = \mathrm{Min}_{z \in X}(v2(xz)+v(zy))$$

Opération de Base (Entête)

- Algorithme OpMat
- Données :
  - M, N : deux matrices d'entiers indicées par les sommets
- Résultat :
  - Res : Une matrice d'entiers indicées par les sommets

Opération de Base (code)

```
DébutCode
    Pour tout x ϵ X faire
        Pour tout y ϵ X faire
            Res[x,y] ← M[x,y] +N[y,y]
            Pour tout z ϵ X faire
                Res[x,y] ← Min(Res[x,y], M[x,z] + N[z,y])
            FinPour
        FinPour
    FinPour
FinCode
```

Complexité

- $O(n^3)$ opérations

Algorithme des chemins de poids min : opération matrice (entête)

- Algorithme CPMOMOM
  - Donnée :
    - M la matrice d'un graphe pondéré
  - Résultat :
    - Res une matrice
    - \\ Res[i,j] est le poids mini d'un chemin entre i et j
  - Variables
    - i entier
    - Inter, Inter2 : deux matrices

Algorithme des chemins de poids min : opération matrice (code)

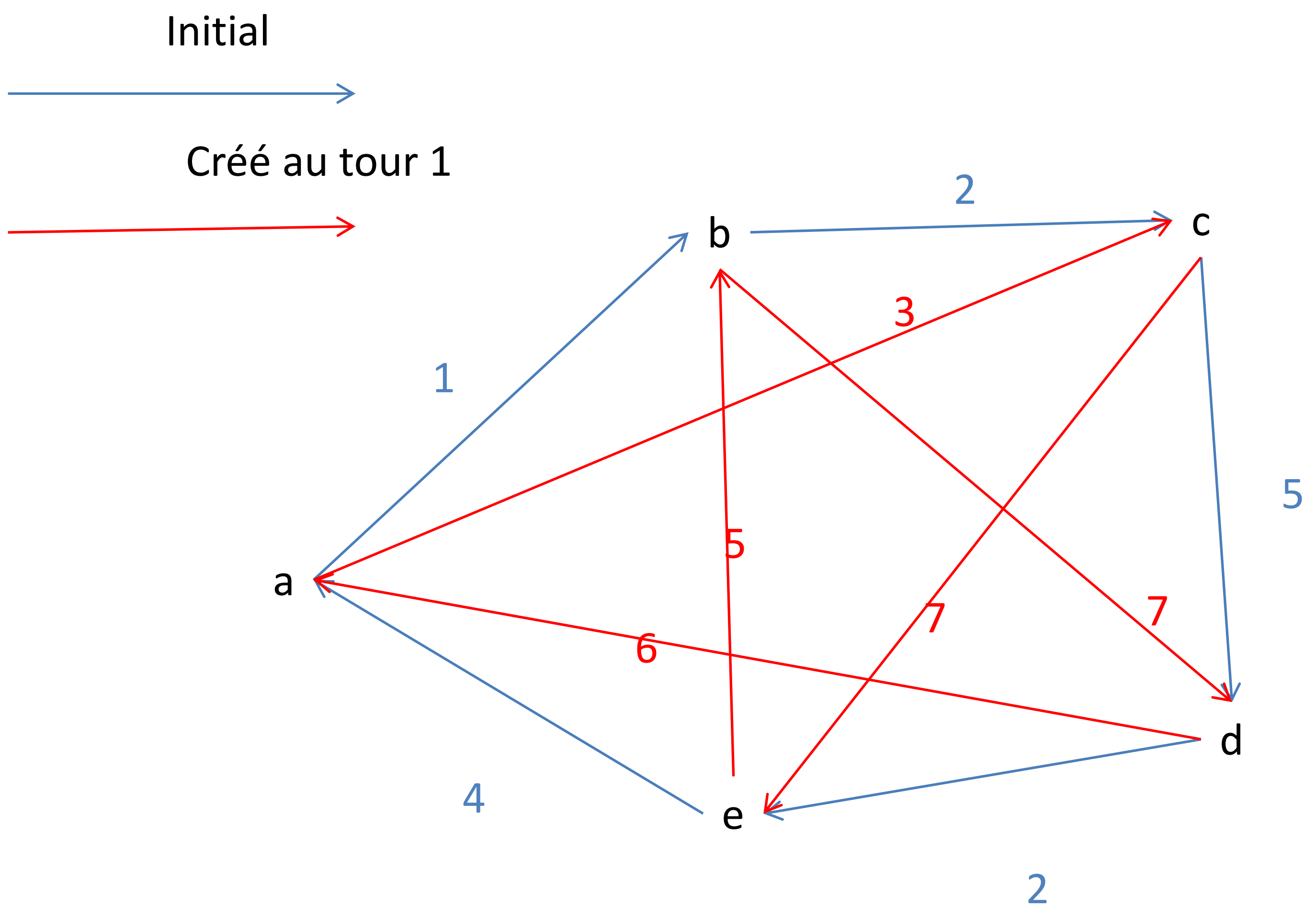
- DébutCode
  - Inter ← M; Inter2← M;// par duplication
  - Pour tout x ϵ X faire
    - Inter[x,x]← 0; Inter2[x,x] ← 0;
  - FinPour
  - Pour i ← 2 à n faire
    - OpMat(Inter, Inter2, Res); Inter ← Res;//Duplication
  - FinPour
- FinCode

Propriétés

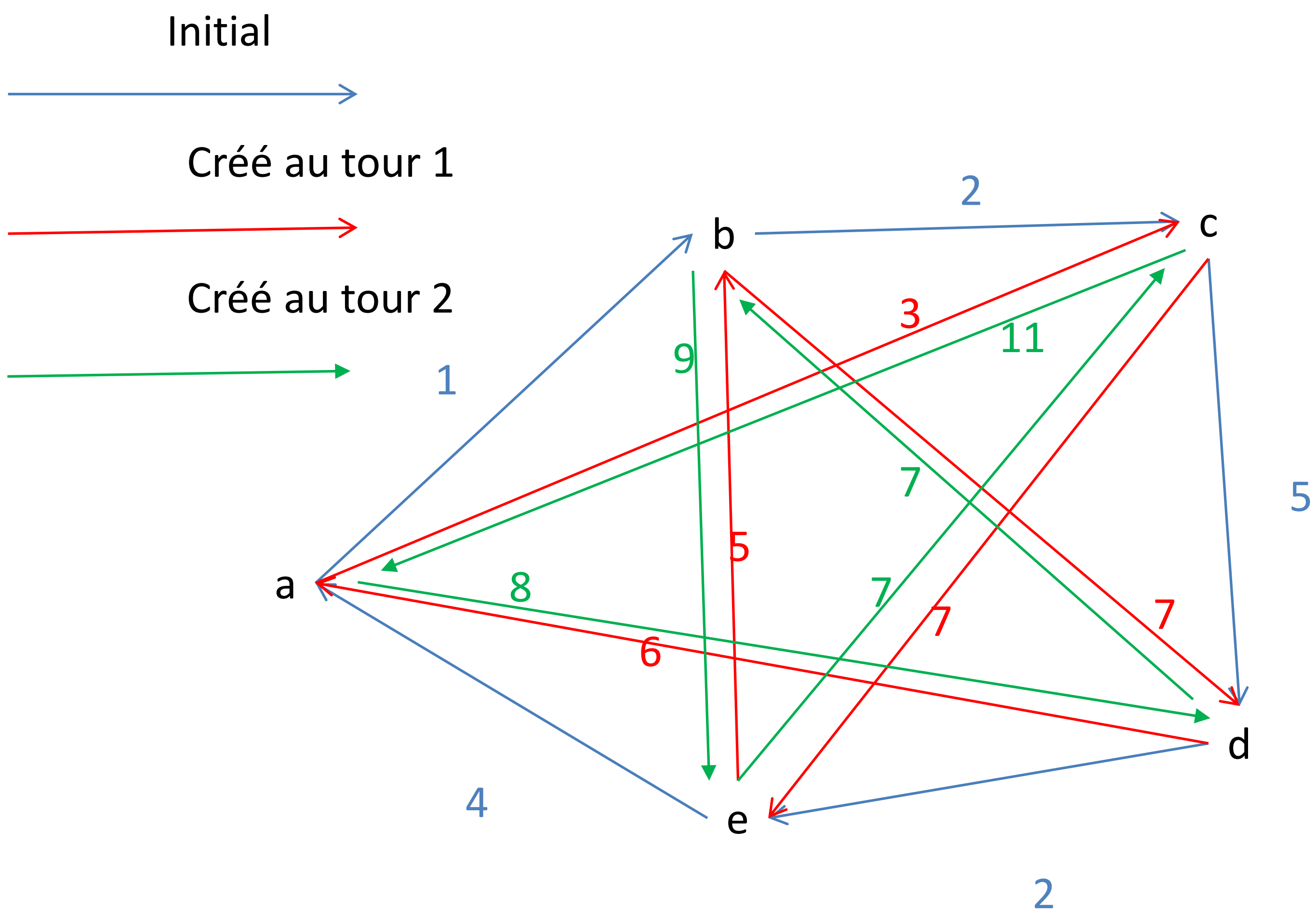
- A la fin du kème tour de boucle, Inter[u,v] contient le poids d’un chemin de u à v de poids minimal et de longueur au plus k+1. C’est l’invariant de l’algorithme

- Complexité $O(n^4)$

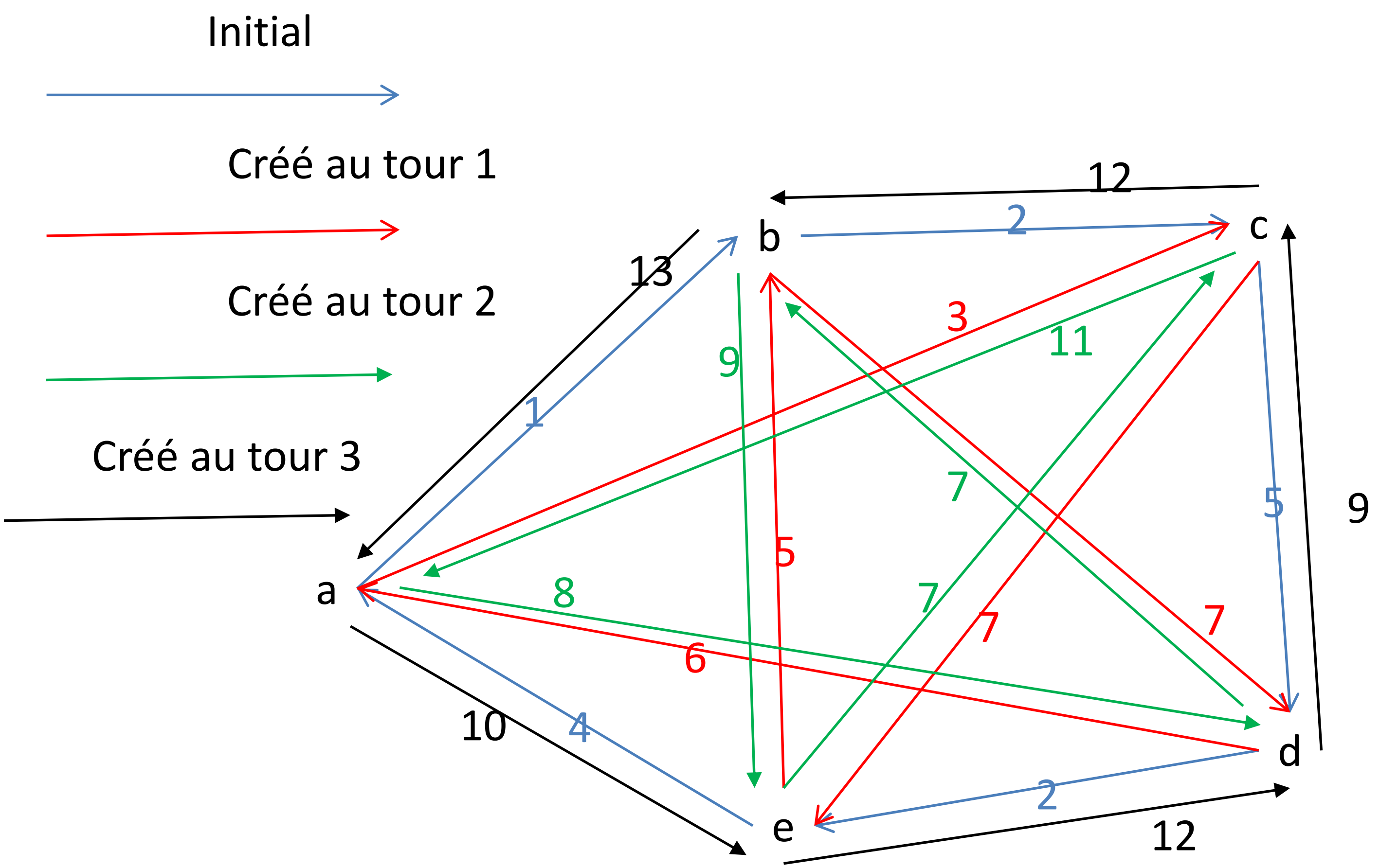
Exemple



Exemple


Initial
Créé au tour 1
Créé au tour 2
a
b
c
d
e
2
1
9
3
11
7
5
5
8
6
7
7
7
4
2

Exemple


Initial
Créé au tour 1
Créé au tour 2
Créé au tour 3
a
b
c
d
e
12
2
13
1
3
11
9
7
5
9
5
8
7
7
7
6
10
4
2
12

Par opérations sur les matrices

- Peut-on aller plus vite ?
  - Si on souhaite calculer le poids de chemins de longueur au plus 4, on peut composer deux chemins de longueurau plus 2 :

$$v4(xy) = Min_{z \in X}(v2(xz)+v2(zy))$$

  - Longueur 8 : $v8(xy) = Min_{z \in X}(v4(xz)+v4(zy))$
  - Longueur 16 : $v16(xy) = Min_{z \in X}(v8(xz)+v8(zy))$
  - Etc

Algorithme des plus courts chemins : opération matrice

- Peut-on aller plus vite ?
  - Or v2 s’obtient par OpMat (M,M)
  - V4 par OpMat(OpMat (M,M), OpMat (M,M))
  - Etc

Algorithme des plus courts chemins : opération matrice (entête)

- Algorithme CPMOMOMV2
  - Donnée :
    - M la matrice d’un graphe pondéré
  - Résultat :
    - Res une matrice
    - \\ Res[i,j] est le poids mini d’un chemin entre i et j
  - Variables
    - i entier
    - Inter, Inter2 : deux matrices

Algorithme des plus courts chemins : opération matrice (code)

```
DébutCode
    i ← 1; Inter ← M; Inter2← M;// par duplication
    Pour tout x ϵ X faire
        Inter[x,x]← 0; Inter2[x,x] ← 0;
    FinPour
    TantQue i < n faire
        OpMat(Inter, Inter2, Res); i ←2*i;
        Inter ← Res; Inter2 ← Res;//Duplication
    FinPour
FinCode
```

Propriétés

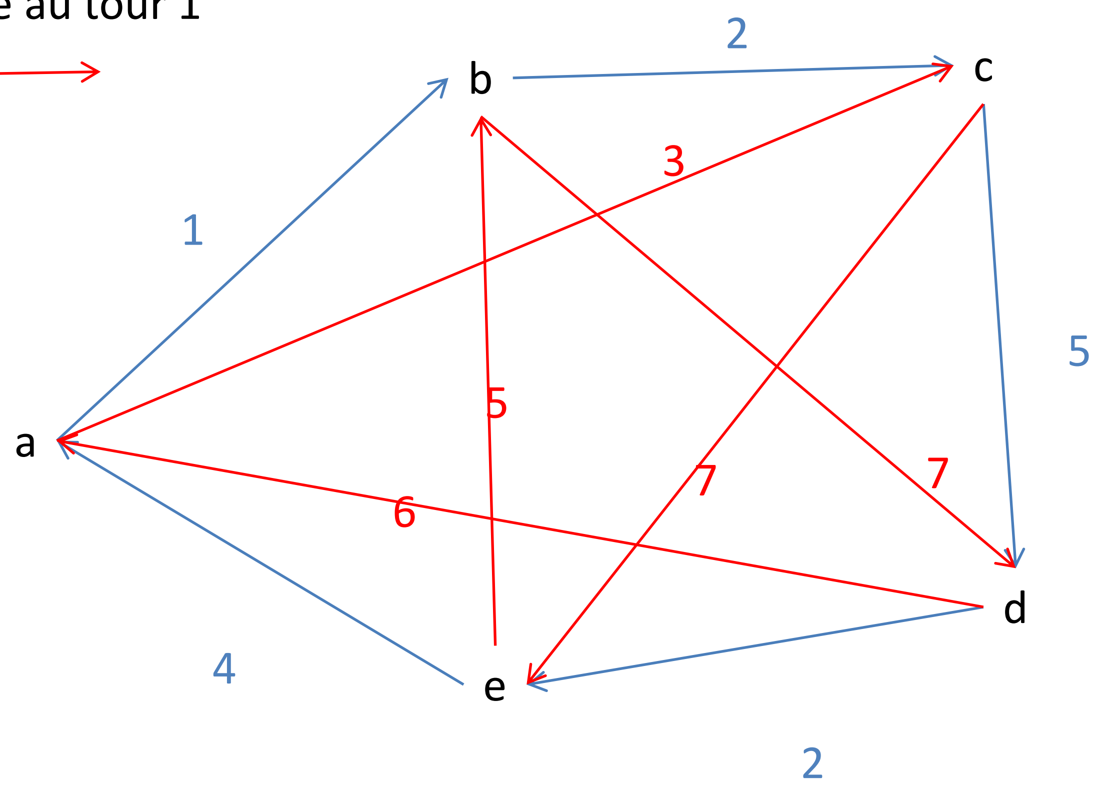
- A la fin du $k^{ième}$ tour de boucle, Inter[u,v] contient le poids d'un chemin de u à v de poids minimal et de longueur au plus 2^k. C'est l'invariant de l'algorithme

- Complexité $O(n^3\log(n))$

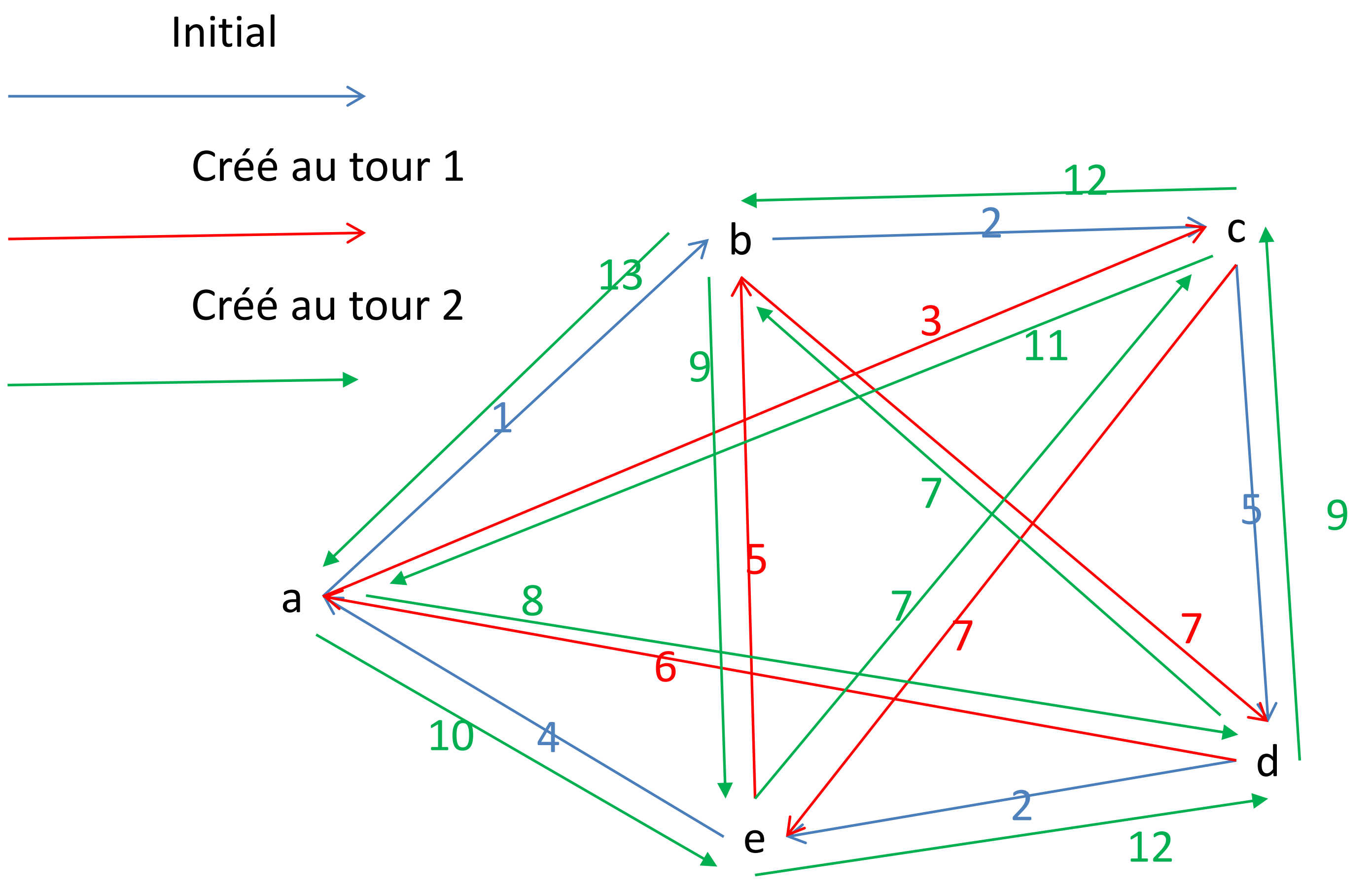
Exemple

Initial

Créé au tour 1



Exemple


Initial
Créé au tour 1
Créé au tour 2
a
b
c
d
e
12
2
13
1
9
3
11
7
5
5
9
8
7
7
7
6
10
4
2
12

Algorithme de Floyd Warshall

- On cherche les chemins de poids minimaux entre tout couple de sommets
- On examine les sommets du graphe un par un dans l'ordre

Principe de l’algorithme

- Pour un couple de sommet (i, j) on peut affirmer que le chemin μ de poids minimal allant de i à j ne passant que par des sommets d’étiquette entre 1 et k
  - soit μ n'emprunte pas le sommet k ;
  - soit μ emprunte exactement une fois le sommet k (car les circuits sont de poids positifs ou nuls) et p est donc la concaténation de deux chemins, entre i et k et k et j respectivement, dont les sommets intermédiaires sont dans {1, 2, 3, …, k-1}.

Algorithme de Floyd Warshall (Entête)

- Algorithme FW
  - Donnée/Résultat :
    - G : Un graphe sous forme de matrice
  - Variables
    - i, j, k : sommets

Algorithme de Floyd Warshall

- DebutCode
  - Pour k ← 1 à n faire M[k,k] ← 0 FinPour
  - Pour k ← 1 à n faire
    - Pour i ← 1 à n faire
      - Pour j ← 1 à n faire
        - » M[i,j] ← Min (M[i,j], M[i,k] + M[k,j])
      - FinPour
    - FinPour
  - FinPour
- FinCode

Algorithme de Floyd Warshall

- Complexité $O(n^3)$

Exemple