Algorithme de Bellman-Ford

- supp les cycles non neg (boucles)
- et mt à jour les vals du graph

Entrée : graphe $G=(V,E)$ avec pondération $\ell \in \mathbb{R}^n$ un sommets (suppose pas cycle neg)

Sortie : Distances de S aux autres sommet

~~[illegible]~~

$D[s] \leftarrow 0$

pour tous les $u \in V \setminus \{s\}$

  $D[u] \leftarrow \infty$

répéter $|V|-1$ fois :

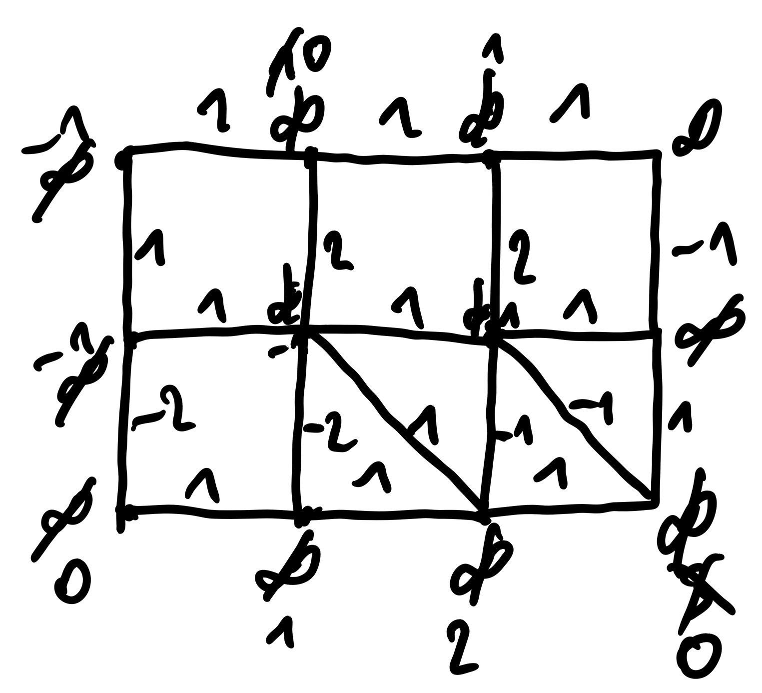
  pour tous les $e \in E$:

    maj(e)

retourner D

procédure maj(u,v)

  $D[v] \leftarrow \min\{D[v], D[u] + \ell(u,v)\}$



procédure maj-prev(u,v) :

  si $D[u] + \ell(u,v) < D[v]$:

    $D[v] \leftarrow D[u] + \ell(u,v)$

    prev[v] ← u

Pour prouver la correction de B-F, il suffit de prouver le lemme suivant.

Lemme : Après i itérations :

- Si $D[u] \neq +\infty$, alors $\exists$ un chemin de s à u de longueur $D[u]$
- s'il $\exists$ un chemin de s à u comprenat au plus i arcs, alors la valeur de $D[u]$ est inférieure ou égale à la longueur d'un plus court chemin de s à u comprenat au plus i arcs.

Complexité de B-F : $O(m \cdot n)$

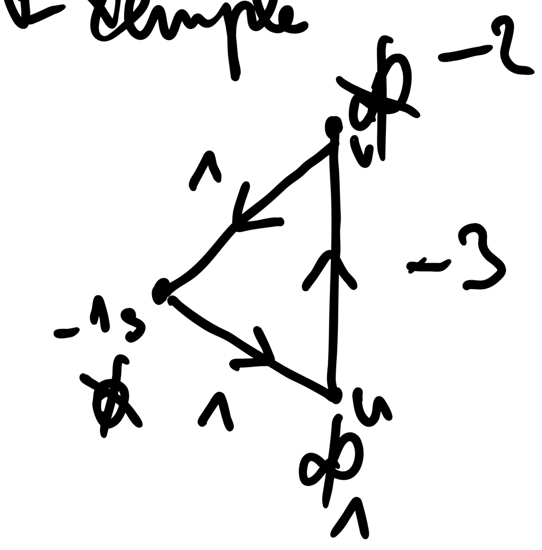
pour les graphes denses (avec $\Omega(n^2)$ arcs) la complexité est de $O(n^3)$

Détection de cycle négatifs :

Un cycle négatif existe dans G ssi il y a au moins 1 changement dans le

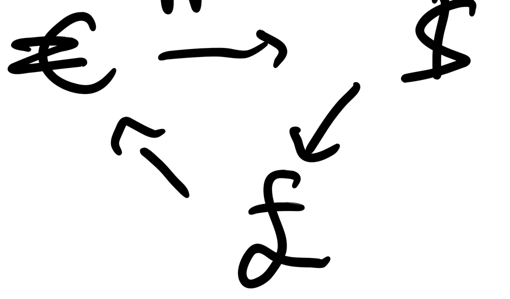
Tableau D après une itération supplémentaire de B-F

Exemple



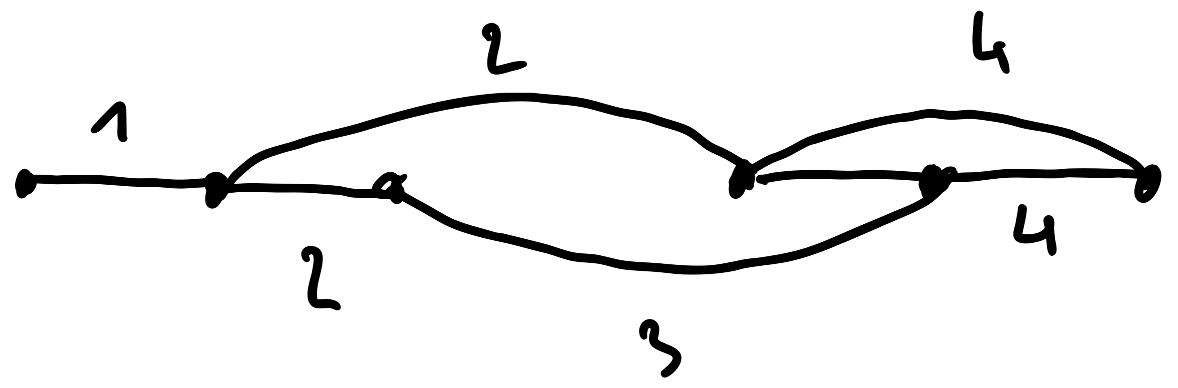
	itération			cycle supp pour vérifier.
sommet	0	1	2	3
S	0	0	0	-1
u	∞	1	1	1
v	∞	∞	-2	-2

ex application taux change



utilisé en finance

Deux classes naturelles de graphes sans cycle négatif :
- les graphes sans arcs négatifs [utiliser Dijkstra]
- les DAG ⟵ y a-t-il un algo plus efficace que Bellman-Ford ?
  ↳ oui! utiliser le tri topologique



Algo de plus court chemin dans les DAG

Entrée : Graphe orienté $G = (V,E)$ avec pondération $l \in \mathbb{R}^n$ un sommet s

Sortie : Distances de s aux autres sommets

$D[S] \leftarrow 0$

pour tous les $u \in V \setminus \{s\}$

$D[u] \leftarrow +\infty$

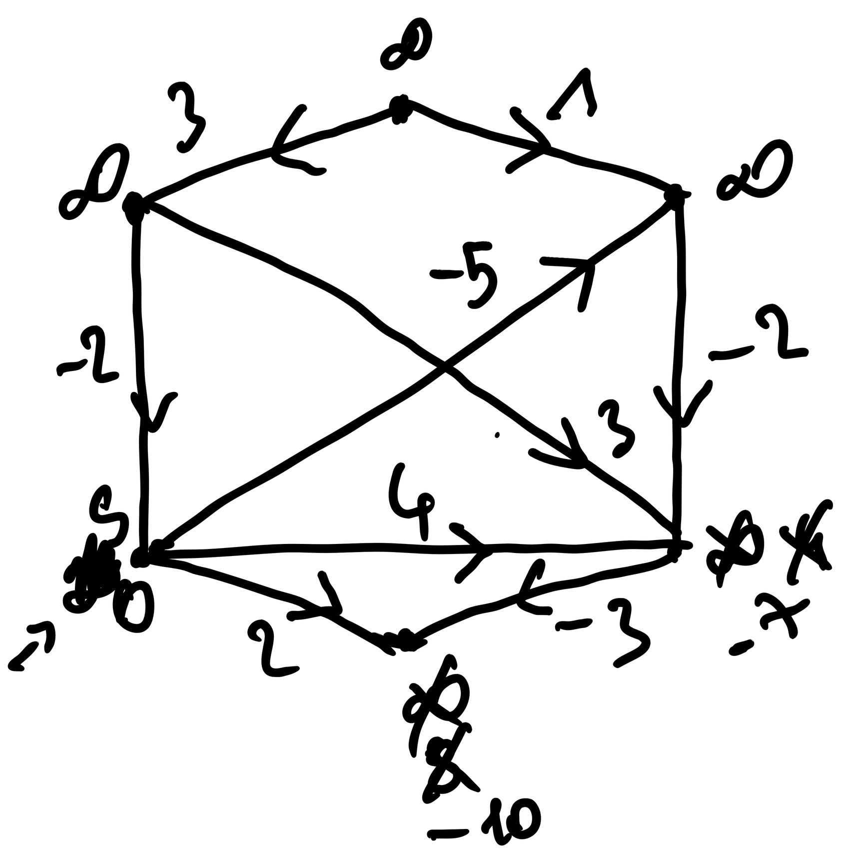
tri topologique de G

pour tous les $u \in V$ dans l'ordre topo :

pour tous les $(u,v) \in E$:

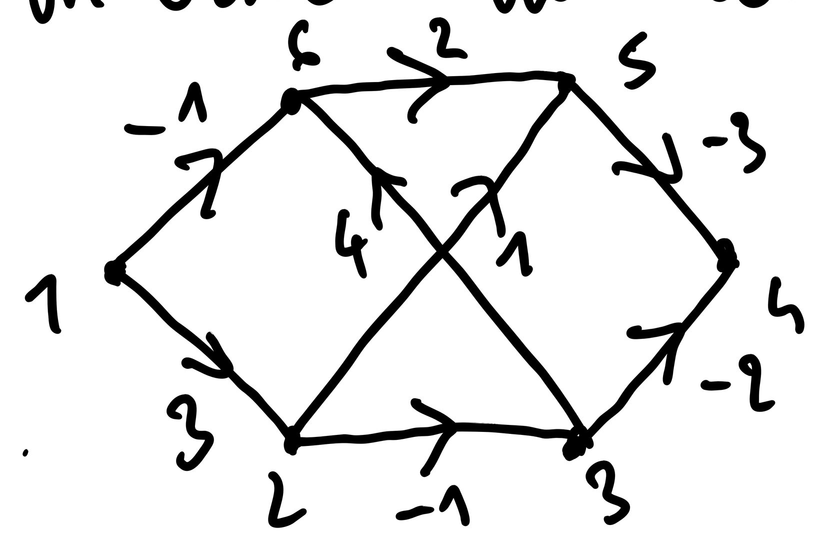
retourner D

Illustration



Distances entre toutes les paires de sommets

On peut utiliser Dijkstra (si pas d'arc négatif) ou B-F n fois, 1 fois pour chaque sommet. La complexité est alors multipliée par n.
En particulier, si le graphe est dense et contient des arcs négatifs, on obtient une complexité de $O(n^4)$



$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 3 & \infty & \infty & \infty & -1 \\ 2 & & 0 & & & & \\ 3 & & & 0 & & & \\ 4 & & & & 0 & & \\ 5 & & & & & 0 & \\ 6 & & & & & & 0 \end{array}$$

Soit $V = \{1, 2, 3, \ldots n\}$ l'ensemble des sommets
Soit $dist(i,j,k)$ = longueur minimum d'un chemin de i à j qui ne passe pas par les sommets $k+1, k+2, \ldots, n$

En particulier :

$$dist(i,j,0) = \begin{cases} l_{i,j} & \text{si } (i,j) \in E \\ +\infty & \text{si } (i,j) \notin E \end{cases}$$

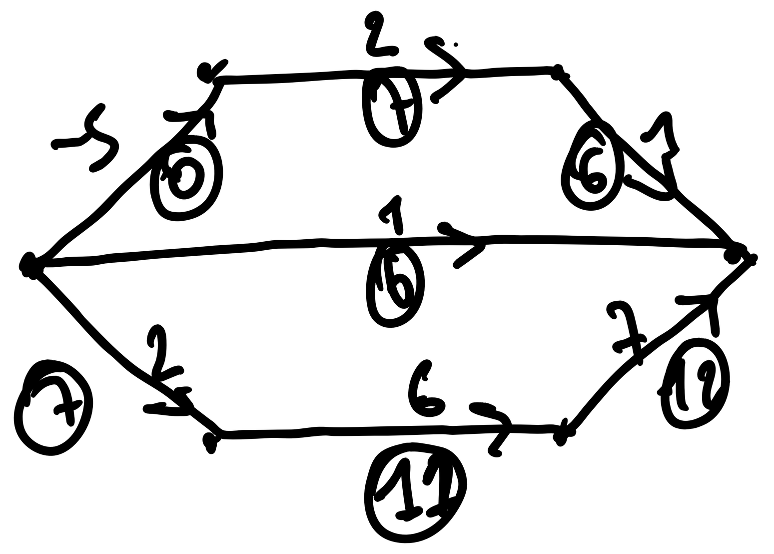
Algorithme de Floyd-Warshall

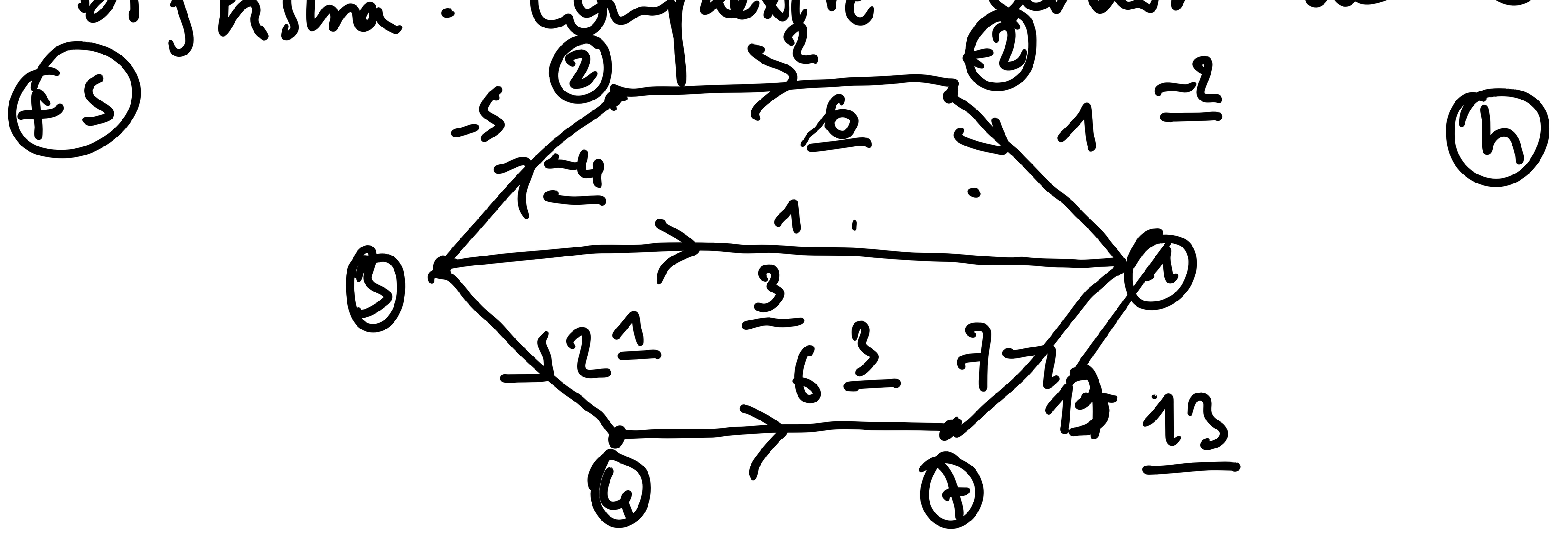
Entrée : Graphe $G = (V, E)$ où $V = \{1, 2, \ldots n\}$, pondération $l \in \mathbb{R}^n$
Sortie : Distances entre chaque paire de sommets

pour tous les $i \in V$:
  pour tous les $j \in V$:
    $dist(i,j,0) \leftarrow +\infty$ ← on met 0 sur les diagonales si pas de boucles.
  pour tous $(i,j) \in E$:
    $dist(i,j,0) \leftarrow l(i,j)$
  pour tous les $k \in V$:
    pour tous les $i \in V$:
      pour tous les $j \in V$:
        $dist(i,j,k) \leftarrow \min\{dist(i,k,k-1) + dist(k,j,k-1), dist(i,j,k-1)\}$

Il est possible de détecter les cycles neg avec Floyd-Warshall (si nbr neg sur la diagonale.

Idée naïve : transformer les poids sur les arcs de sorte que tous les arcs sont non négatifs et ensuite utiliser Dijkstra. Complexité serait de $O(nm + n^2 \log n)$





Repondération des arcs qui préserve les plus courts chemins

- Soit $G = (V, E)$ un graphe avec pondération $l \in \mathbb{R}^n$
- Soit $h \in \mathbb{R}^n$ un vecteur associant à chaque sommet un nombre réel.
- On définit une nouvelle pondération $l' \in \mathbb{R}^m$ par

$$l'(u,v) = l(u,v) + h(u) - h(v)$$

Lemme : P est un plus court chemin de u à v par rapport à l

$$\Updownarrow$$

P est un plus court chemin de u à v par rapport à l'

Preuve : Soit P un chemin quelconque de u à v

$$l'(P) = \sum_{i=1}^{k} l'(v_{i-1}, v_i)$$

$$= \sum_{i=1}^{k} \left(l(v_{i-1}, v_i) + h(v_{i-1}) - h(v_i) \right)$$

$$= \sum_{i=1}^{k} l(v_{i-1}, v_i) + h(u) - h(v) = l(P) + h(u) - h(v)$$

En particulier, si P est un chemin de u à v, alors $l(P) = \mathrm{dist}_l(u,v)$ ssi $l(P') = \mathrm{dist}_{l'}(u,v)$

Remarque : la longueur des cycles ne change pas