BELMANN - FORD

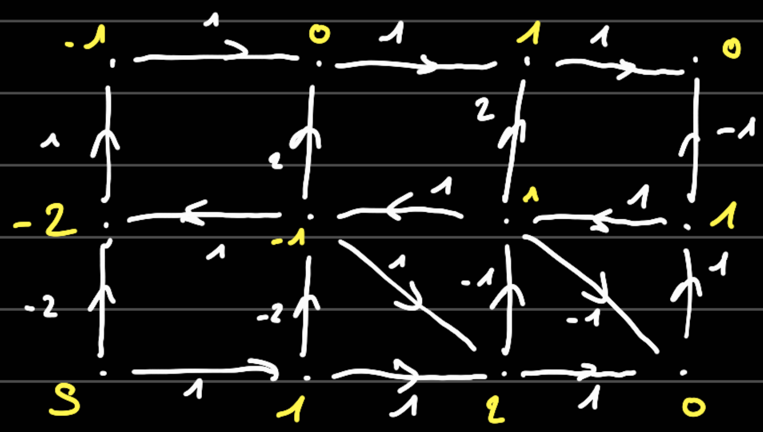
<u>Entrée</u> : $G = (V, E)$ avec pondération
$\ell \in \mathbb{R}^n$

un sommet s

<u>Sortie</u> : Distance de s aux autres sommets

$D[s] \leftarrow 0$ ← tableau des distances
pour tout $v \in V \setminus \{s\}$:
  $D[v] \leftarrow \infty$
repeter $|V| - 1$ 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 $[v] \leftarrow u$

Lemme : Après i itérations.

- Si $D[u] \neq \infty$,
  alors il existe un chemin de s à u de longueur $D[u]$.

- S'il existe un chemin de s à u comprenant ou plus i arcs,
  alors la valeur de $D[u]$ inf ou ég. à la longueur d'un plus court chemin de s à u avec au plus i arcs.

Complexité : $O(m.n)$

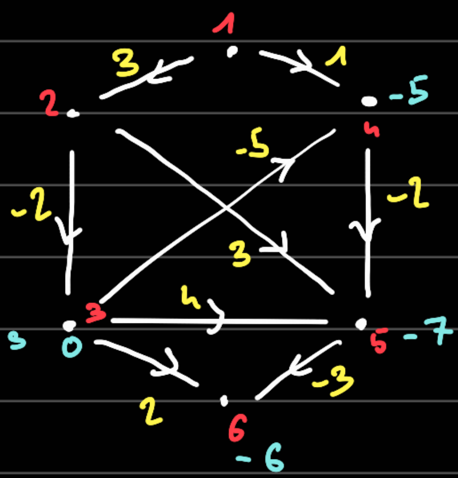
Pour les graphes denses ($\Omega(n^2)$ arcs)
alors $O(n^3)$

Il existe des cycles négatifs dans G
ssi il y a moins 1 chgmt ds le tableau D
après une itération supplémentaire de B-F

Deux classes naturelles de graphes sans cycle négatif :

- Les graphes sans arcs négatifs
  => Dijkstra
- Les DAG ← qqch de plus efficace que B.F ?
  => Tri topologique

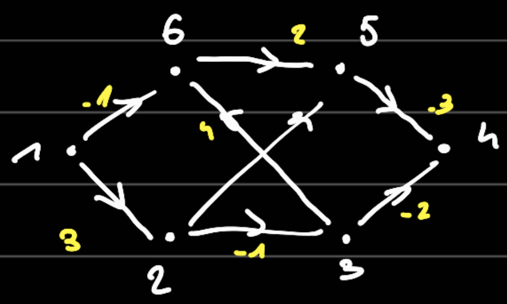
Illustration :



↳ Distances entre toutes les paires de sommets ?

On peut utiliser Dijkstra (si il n'y a pas d'arc négatif) ou B-F sinon.
Dans tous les cas, on l'utilise n fois.

=> Problème si le graphe est dense car on a $O(n^4)$



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

(on remplace les 0,1 par les poids vers le sommets)

Soit $V = \{1, 2, \dots, n\}$ l'ensemble des sommets

Soit $dist(i, j, k)$ = longueur minimum d'un chemin i à j qui passe par les sommets $k+1, k+2, \dots, n$.

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

Algorithme de Floyd - Warshell :

Entrée: $G = (V, E)$ où $V = \{1, 2, 3, \dots, n\}$
pondération $\ell \in \mathbb{R}^n$

Sortie: Distances entre chaque paire de sommets pour tout $i \in V$

  pour tout $j \in V$

    $dist(i,j,0) \leftarrow \infty$

pour tout $(i,j) \in E$:

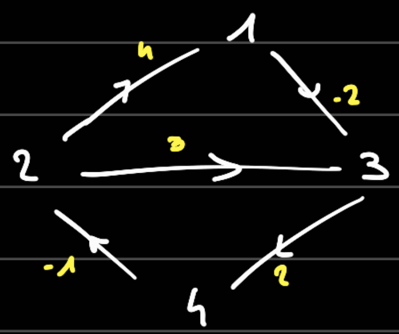
  $dist(i,j,0) \leftarrow \ell(i,j)$

pour tout les $k \in V$:

  pour tous les $i \in V$

    pour tous les $j \in V$:

      $dist(i,j,k) \leftarrow \min \{ dist(i,k,k-1) + dist(k,j,k-1),\ dist(i,j,k-1) \}$



	1	2	3	4
1	0	∞	-2	∞
2	4	0	3	∞
3	∞	∞	0	2
4	∞	-1	∞	0

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

$\Updownarrow$

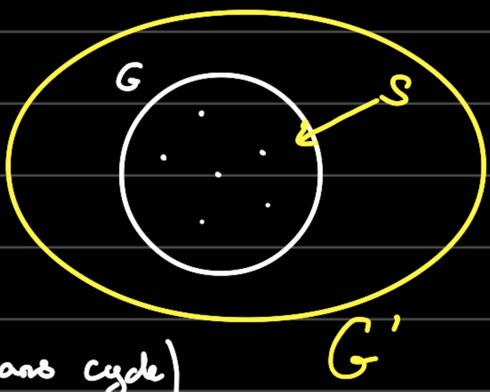
P est un plus court chemin de u à v par rapport à ℓ'.

où $\ell'(u,v) = \ell(u,v) + h(u) - h(v)$

En particulier : si $\nexists$ cycle neg avec ℓ

$\Rightarrow \nexists$ ______ ℓ'

↳ Comment trouver une fonction
$h : V(G) \to \mathbb{R}$
tq il n'y a pas d'arêtes négatives par rapport à ℓ' ?



(G sans cycle négatif)

Création d'un point s relié à tous les sommets de G. Le graphe ainsi créé s'appelle G', et et $\{\forall v \in V(G) \ \mathrm{arc}(s, v)\}$ l'ensemble des arcs créés.

Remarque : G' ne contient pas de cycle négatif car G n'a pas de cycle négatif.
(car $\forall sv \in E(G)$ sont des arcs sortants)

On définit $h(v)$ comme la distance de s à v dans G'
$\to \quad h(v) = \mathrm{dist}_{G'}(s, v)$

On a $h(v) \leqslant h(u) + \ell(u,v)$
$\forall uv \in E(G)$ par Belmann-Ford

on sait que $\ell'(u,v) = \ell(u,v) + h(u) - h(v)$
donc $\ell'(u,v) \geqslant h(v) - h(u) + h(u) - h(v)$
$\ell'(u,v) \geqslant 0$

Donc on pourra utiliser Dijkstra.

Algorithme de Johnson

1) Calculer G' (ajouter s et $arc(s, v)\ \forall v \in V(G)$)
2) Appliquer Bellman-Ford à G', avec source $= s$ pour calculer $h(v) = dist_{G'}(s, v)$
3) Repondérer chaque arc $(u,v) \in E(G)$ par $\ell'(u,v) = \ell(u,v) + h(u) - h(v) \geq 0$
4) $\forall u \in V(G)$, exécuter Dijkstra pour calculer $dist(u,v)\ \forall v \in V(G)$
5) $\forall (u,v) \in V(G)^2$, $dist_\ell(u,v) = dist_{\ell'}(u,v) - h(u) + h(v)$

Complexité de Johnson :

(par les étapes notées précédemment)

1) $O(n)$
2) $O(nm)$
3) $O(m)$
4) Dijkstra : $O(m + n\log(n))$
   qu'on fait s/ tous les sommets
   $\Rightarrow O(nm + n^2\log(n))$
5) $O(n^2)$

Au final la complexité de Johnson est un $O(nm + n^2\log(n))$

- Si le graphe est dense (m est proche de n $\Leftrightarrow \Omega(n^2)$ arêtes)

alors complexité de Johnson $= O(n^3)$
(car $m \in \Omega(n^2)$ donc $m.n \in \Omega(n^3) > O(n^2\log(n))$)

- Sinon Johnson est plus efficace que Floyd-Warshall.