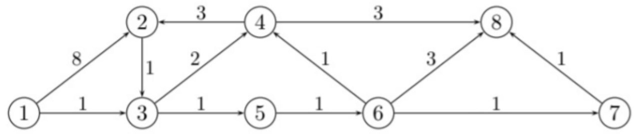
Exercice 37 (Algorithme de Dijkstra)

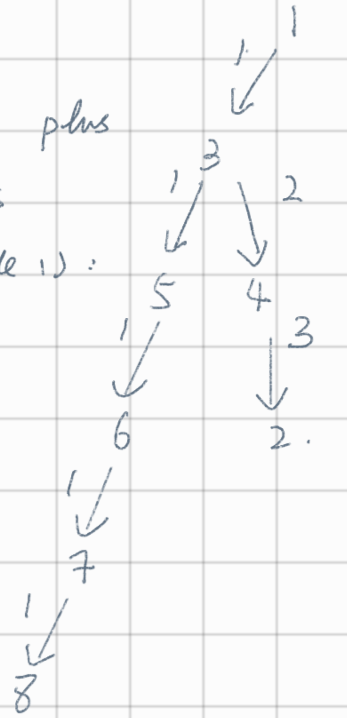
Q 37.1 Appliquer l'algorithme de Dijkstra pour obtenir les plus courts chemins de 1 à tous les autres sommets sur le graphe suivant :

所有 arcs ≥ 0. → 可用 Dijkstra



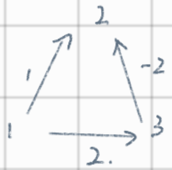
	1	2	3	4	5	6	7	8
λ / p	0 / –	∞ / –	∞ / –	∞ / –	∞ / –	∞ / –	∞ / –	∞ / –
		8 / 1	$\boxed{1 / 1}$	∞ / –	∞ / –	∞ / –	∞ / –	∞ / –
		8 / 1		3 / 3	$\boxed{2 / 3}$	∞ / –	∞ / –	∞ / –
		8 / 1		$\boxed{3 / 3}$		3 / 5	∞ / –	∞ / –
		6 / 4				$\boxed{3 / 5}$	∞ / –	6 / 4
		6 / 4					$\boxed{4 / 6}$	6 / 4
		6 / 4						$\boxed{5 / 7}$
		$\boxed{6 / 4}$						

Aborscence du plus courts chemins (à partir de 1) :



Q 37.2 Trouver une instance de graphe qui a des arcs de longueurs négatives et ne possède pas de circuit négatif, sur laquelle l'algorithme de Dijkstra ne donne pas satisfaction.

Si il y a des arcs de longueur < 0, l'algo Dijkstra peut donner une mauvais réponse dans un graphe sans circuit.



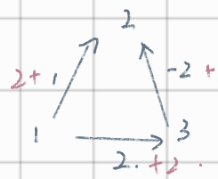
1	2	3
$\boxed{0 / –}$	∞ / –	∞ / –
	$\boxed{1 / 1}$	2 / 1
		2 / 1

PCCH de 1 à 2 de longueur 1

Mais opt = 2 − 2 = 0.

Q 37.3 Un ingénieur suggère la méthode suivante pour trouver le plus court chemin d'un sommet s à un sommet t dans un graphe orienté avec des arcs de valeurs négatives : ajouter une grande constante à chaque arête de façon à ce que toutes les valeurs deviennent positives, faire tourner l'algorithme de Dijkstra depuis le sommet s, et retourner le plus court chemin trouvé au sommet t. Cette méthode est-elle valide ? Prouver la validité ou fournir un contre-exemple dans le cas contraire.

Ne marche pas.

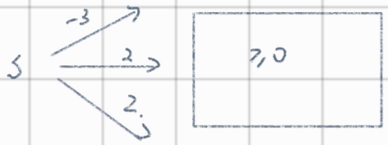


1	2	3
$\boxed{0 / –}$	∞ / –	∞ / –
	$\boxed{3 / 1}$	4 / 1
		$\boxed{4 / 1}$

Mais opt = 1 → 3 → 2 = 0.

Marche si tous les chemins de s à t avait m̂ nb d'arcs.

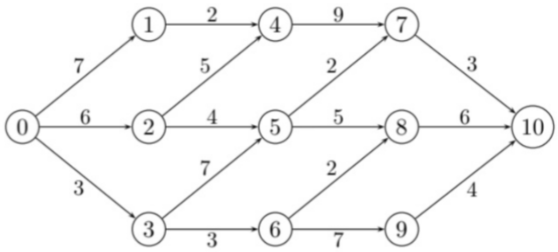
Q 37.4 On considère maintenant un graphe orienté dans lequel les seules arêtes négatives sont celles issues du sommet s de départ. Est-ce que l'algorithme de Dijkstra peut fournir un résultat non-valide sur un tel graphe ? Prouvez votre réponse.



Oui. Pas de circuit absorbants.
Pas d'arc entrant en s.

Exercice 38 (Algorithme de Bellman)

Considérons le projet de construction d'une autoroute entre les villes 0 et 10. Les arcs représentent les différents tronçons possibles de l'autoroute. Chaque arc est valué par le coût total de réalisation du tronçon correspondant.

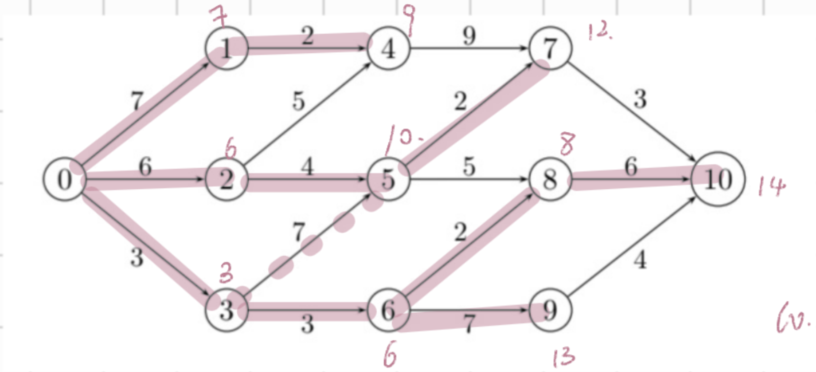


Q 38.1 En utilisant l'algorithme de Bellman, déterminer le tracé dont le coût total de construction est minimum et celui dont le coût total est maximum.

Bellman : graphe sans circuit (ok avec longueur < 0).

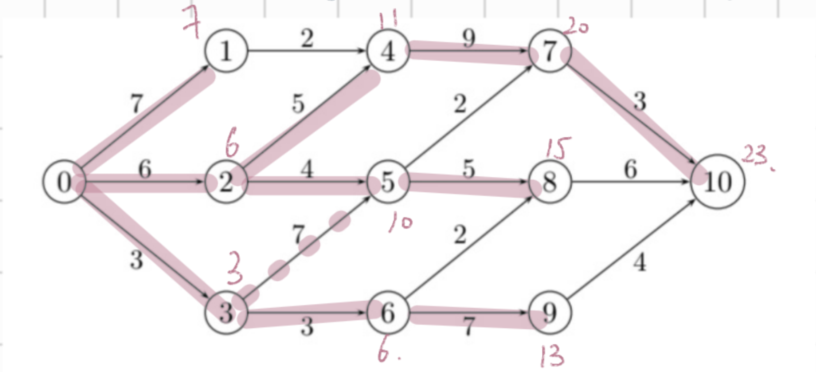
ordre topologique : $1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7 \to 8 \to 9 \to 10$.

* $\lambda(j) = \min \{\lambda(i) + d(i,j) : i \text{ prédent de } j\}$



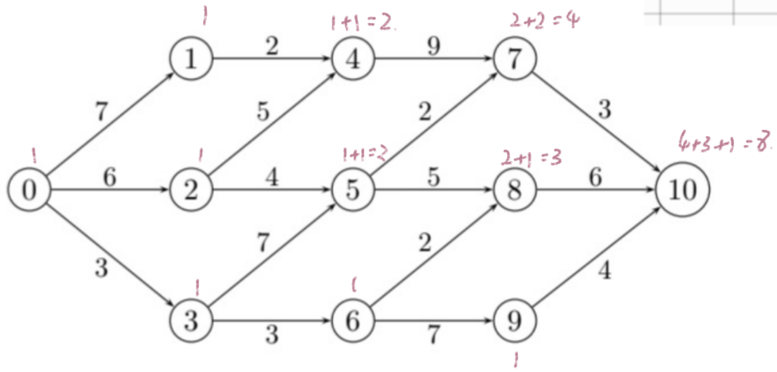
(0. 3. 6. 8. 10) de longueur 14.

* Max $\begin{cases} \lambda(0) = 0. \\ \lambda(j) = \max \{\lambda(i) + d(i,j) : i \text{ précédent de } j\} \end{cases}$



$0 \to 2 \to 4 \to 7 \to 10$: 23.

Q 38.2 Indiquer comment modifier l'algorithme si l'on souhaite maintenant connaître le nombre de tracés possibles.



$\lambda(0) = 1$

$\lambda(j) = \sum_{i \text{ pred de } j} \lambda(i)$

partit° des chemins de 0 à j

ie. f° du dernier sommet avant j

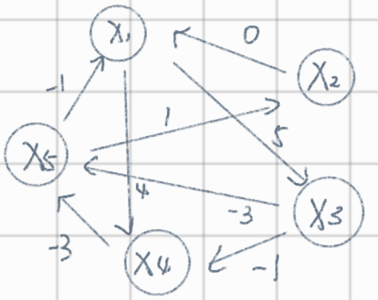
总结:
- arc de poids 1 : BFS
- graphe sans circuit : Bellman
- arc de poids ≥ 0 : Dijkstra
- Sinon : Bellman - Ford

Exercice 43 (Système de différences)

Trouver une solution réalisable pour le système suivant ou déterminer qu'une telle solution n'existe pas.

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 0 \\ -1 \\ 1 \\ 5 \\ 4 \\ -1 \\ -3 \\ -3 \end{pmatrix}$$

$$\begin{cases} x_1 - x_2 \leq 0 \\ x_1 - x_5 \leq -1 \\ x_2 - x_5 \leq 1 \\ -x_1 + x_3 \leq 5 \\ -x_1 + x_4 \leq 4 \\ -x_3 + x_4 \leq -1 \\ -x_3 + x_5 \leq -3 \\ -x_4 + x_5 \leq -3 \end{cases}$$



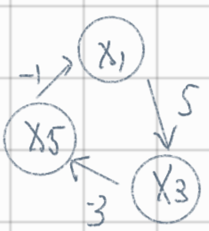
On pourrait utiliser PL.

Ici graphe (équat° $x_j - x_i \leq \lambda$)

↑ arc de x_i à x_j de poids λ.

On met - un sommet par variable.

- un arc $x_i \to x_j$ de longueur λ par chaque inégalité $x_i - x_j \leq \lambda$.



$x_3 - x_1 \leq 5$

$-x_3 + x_5 \leq -3$

$x_1 - x_5 \leq -1$

$\Rightarrow \quad 0 \leq 1$

负数环,

若为 $0 \leq -1$ ⇒ Arbre absorbants

incompatibilité ⇒ pas de solut°

△ si le graphe a un circuit absorbant. ⇒ le système n'a pas de solut°

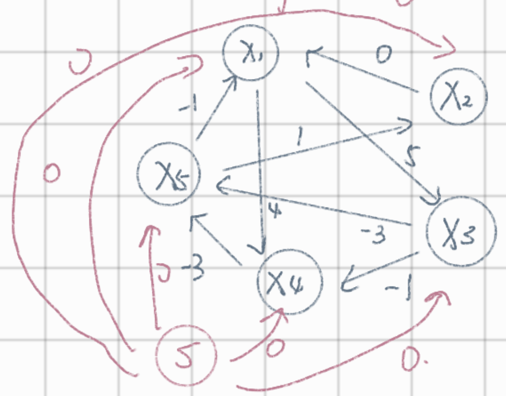
Soit $C = (x_{i_1} \cdots \xrightarrow{\lambda_k} x_{i_k} \; x_{i_1})$ est un circuit absorbant

Cela correspond aux inégalité. $\begin{cases} x_{i_2} - x_{i_1} \le \lambda_1 \\ x_{i_3} - x_{i_2} \le \lambda_2 \\ \vdots \\ x_{i_1} - x_{i_k} \le \lambda_k \end{cases}$ ⇒ $0 \le \lambda_1 + \lambda_2 + \cdots \lambda_k < 0$.

⇒ Contradit°.

On obtient $\lambda_1 + \cdots \lambda_k$ est la longueur de circuit (absorbant) Donc $\sum_{i=1}^{k} \lambda_k < 0$.

Δ. On veut mq le graphe contient un circuit absorbant ⇒ Pas de solut°



Dans le graphe. on ajoute une source s.

et les arcs (s, x_i) de longueur 0.

Si le graphe ne contient pas de circuit absor

soit N_i la valeur d'un PCCH de s à N_i

$x_j - x_i \le \lambda \qquad x_i \to x_j$

on choisit $x_i = u_i$

L'arc (x_i, x_j) de valeur extraire $u_j \le u_i + \lambda$

⇒ Donc $\qquad x_j - s \le x_i - s + \lambda$

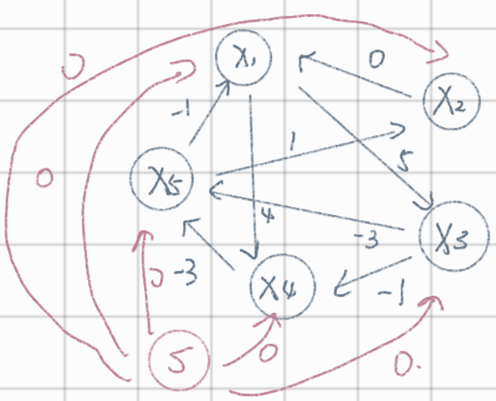
⇒ $\qquad x_j - x_i \le \lambda$

Donc les x_i les solut° du système.

On utilise l'algo de Ford-Bellman ⇒ déterminer si le graphe a un circuit absorbant

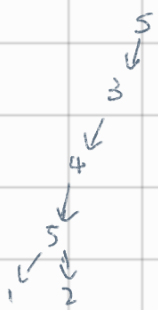
sinon les valeurs de PCCH à partir de s.

$\lambda_{k+1}(x_j) = \min\{\lambda_k(x_j);\ \min\{\lambda_k(x_i) + \ell(x_i, x_j) : i \text{ pred de } j\}\}$



0	1	2	3	4	5
0 / –	∞ / –	∞ / –	∞ / –	∞ / –	∞ / –
0 / –	0 / 5	0 / 5	0 / 5	0 / 5	0 / 5
0 / –	-1 / 5	0 / 5	0 / 5	-1 / 3	-3 / 3
0 / –	-4 / 5	-2 / 5	0 / 5	-1 / 3	-4 / 4
0 / –	-5 / 5	-3 / 5	0 / 5	-1 / 3	-4 / 4
0 / –	-5 / 5	-3 / 5	0 / 5	-1 / 3	-4 / 4

相同 ⇒ 停



Sinon. 有几个 sommet 运行几次.

$x_1 = -5 \quad x_2 = -3 \quad x_3 = 0 \quad x_4 = -1 \quad x_5 = -5.$

Exercice 39 (Conversions de devises)

Un "trader" se trouve devant la possibilité d'intervenir, sans aucun frais fixe, sur le marché des devises en €, \$, £ et ¥. A un moment donné, il observe les taux de change suivants :

↷	€	\$	£	¥
€	–	1.256	0.673	127
\$	0.796	–	0.536	109
£	1.486	1.867	–	203
¥	0.0073	0.092	0.005	–

Disposant d'un capital initial en Euros, le trader cherche à déterminer les séquences d'échanges permettant de faire fructifier au mieux ce montant à un horizon k donné (c'est-à-dire qu'on se donne au plus k échanges au terme desquels on souhaite disposer d'une somme en Euros qui soit la plus grande possible). On suppose que les taux de change sont stables sur la période considérée.

Q 39.1 Définir un graphe G_k sans circuit qui permette de formuler ce problème comme un problème de chemin de valeur maximale, pour un horizon k donné (on précisera ce que représentent les arcs du graphes, les valeurs des arcs, et comment on définit la valeur d'un chemin).

①

$v(p) = \prod_{v \in p} l(v)$

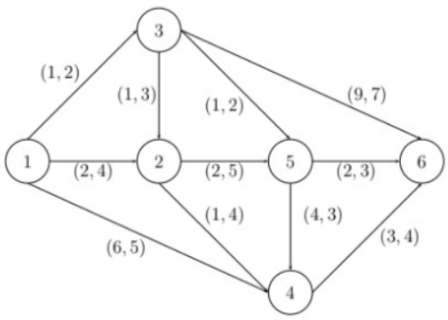
On cherche un chemin de € à € avec au plus k arcs. de valeur v(p) maximale.

C'est λ_k dans l'algo Ford-Bellman.

ou ②.

Exercice 41 (Chemin de qualité/coût optimal)

On considère le graphe ci-dessous à $n = 6$ sommets et $m = 11$ arcs. Il s'agit d'un graphe orienté. A chaque arc u du graphe sont associées deux valeurs numériques c_u (le "coût" de l'arc u) et q_u (la "qualité de service" de l'arc u). Ainsi, pour l'arc $(1, 2)$, $c_{12} = 2$ et $q_{12} = 4$, etc...



Q 41.1 Le graphe ci-dessus contient-il, ou non, un circuit ? Répondre soit en exhibant un circuit, soit en exhibant un *ordre topologique* et en expliquant de façon précise pourquoi l'existence d'un ordre topologique garantit l'absence de circuit.

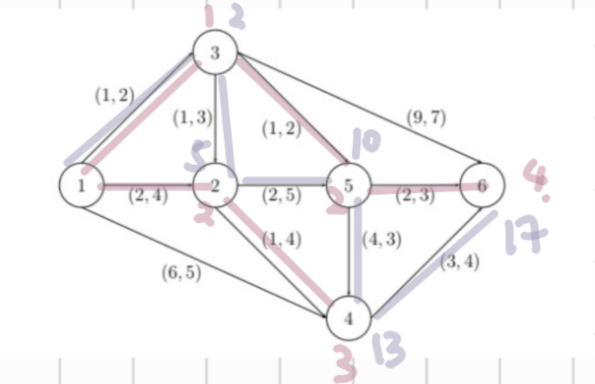
Ordre topologique (1, 3, 2, 5, 4. 6).

Q 41.2 Déterminer un chemin de coût total minimum entre les sommets 1 et 6. On expliquera l'algorithme utilisé et on détaillera toutes les étapes intermédiaires de son fonctionnement sur l'exemple. Quelle est la qualité totale de ce chemin (somme des q_u sur les arcs u du chemin) ? Déterminer le rapport qualité/coût de ce chemin.

Le graphe est sans circuit ⇒ Utiliser Bellman.

Q 41.2 Déterminer un chemin de coût total minimum entre les sommets 1 et 6. On expliquera l'algorithme utilisé et on détaillera toutes les étapes intermédiaires de son fonctionnement sur l'exemple. Quelle est la qualité totale de ce chemin (somme des q_u sur les arcs u du chemin) ? Déterminer le rapport qualité/coût de ce chemin.

Q 41.3 Déterminer un chemin de qualité totale maximale entre les sommets 1 et 6. Détailler les étapes du calcul. Quel est le coût total de ce chemin ? Déterminer le rapport qualité/coût de ce chemin.



(1, 3, 5, 6) $C_{min} = 4$ $Q = 7$ rapport $7/4$

(1, 3, 2, 5, 4, 6) $Q_{max} = 17$ $C = 11$ rapport $17/11$.

Q 41.4 Entre les chemins trouvés à la question 2 et à la question 3, quel est celui qui a le plus grand rapport qualité/coût ? Ce chemin est-il effectivement le meilleur au sens du rapport qualité/coût ? Autrement dit, peut-on trouver un chemin entre 1 et 6 ayant un rapport qualité/coût supérieur ? Si oui, préciser de quel chemin il s'agit.

(1, 3, 5, 6) est meilleur que (1, 3, 2, 5, 4, 6) !

Mais (1, 3, 2, 4, 6) $C = 6$. $Q = 13$. $13/6 > 7/4$

Q 41.5 Soit λ^* la valeur d'un chemin de rapport qualité/coût maximum. Étant donné $\lambda \in \mathbb{R}$, on associe maintenant à chaque arc u du graphe un poids $w_u = q_u - \lambda c_u$. On note $G(\lambda)$ le graphe ainsi valué. Montrer que $\lambda < \lambda^*$ si et seulement si il existe dans $G(\lambda)$ un chemin de valeur strictement positive. Montrer ensuite que $\lambda > \lambda^*$ si et seulement si la valeur de tout chemin dans $G(\lambda)$ est strictement négative. En déduire une condition nécessaire et suffisante pour que $\lambda = \lambda^*$.

• Max $\frac{\sum_v q_v}{\sum_v c_v}$ $\quad w_v^\lambda = q_v - \lambda c_v$.

$w_v^\lambda = q(p) - \lambda c(p) \geq 0 \iff q(p) \geq \lambda c(p)$

$\iff \frac{q(p)}{c(p)} \geq \lambda$.

Idée : Si on trouve 'le bon λ' alors trouver le meilleur chemin selon w_v^λ donne un chemin de $\frac{q}{c}$ optimal

On note p^λ un chemin maximisant $w^\lambda(p) = \sum_v w_v^\lambda$. λ^* $\frac{q(p)}{c(p)}$ opt.

* . Montrons que $\lambda < \lambda^* \iff w^\lambda(p^\lambda) > 0$.

Si $w^\lambda(p^\lambda) > 0$ alors $q(p^\lambda) - \lambda c(p^\lambda) > 0 \iff \frac{q(p^\lambda)}{c(p^\lambda)} > \lambda \iff \lambda^* > \lambda$

Si $\lambda < \lambda^*$. Soit p^*, $\frac{q(p^*)}{c(p^*)} = \lambda^* > \lambda \iff w^\lambda(p^*) = q(p^*) - \lambda c(p^*) > 0$.

Ainsi $w^\lambda(p^\lambda) \geq w^\lambda(p^*) > 0$

$*$ Mq $\lambda > \lambda^* \iff W^\lambda(p^\lambda) < 0$.

$W^\lambda(p^\lambda) < 0 \iff \forall p \; W^\lambda(p) < 0 \iff \forall p. \; g(p) - \lambda c(p) < 0$

$\iff \forall p \; \frac{g(p)}{c(p)} < \lambda \iff \lambda^* < \lambda$

Ainsi $\lambda = \lambda^* \iff W^\lambda(p^\lambda) = 0$.

Q 41.6 Expliciter brièvement le principe général d'un algorithme pour déterminer un chemin de rapport qualité/coût maximum dans un graphe sans circuit bi-valué.

Idée : Trouver M_{min} et M_{max} tq $M_{min} \leq \lambda^* \leq M_{max}$

On procède par dichotomie

$\lambda = \frac{M_{min} + M_{max}}{2}$.

$P\lambda$ si $W^\lambda(p^\lambda) = \begin{cases} 0 & \text{trouvé} \\ >0 & \lambda < \lambda^* \quad M_{min} \leftarrow \frac{M_{min} + M_{max}}{2} \\ <0 & \lambda > \lambda^* \quad M_{max} \leftarrow \frac{M_{min} + M_{max}}{2} \end{cases}$