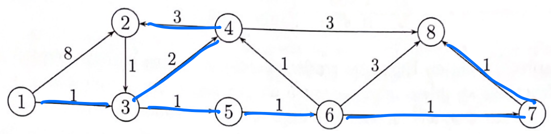
6 Algorithmes de plus court chemin

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 :



arborescence du plus court chemin 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.

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.

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.

Tous les arcs sont de longueur ≥ 0 → on peut utiliser Dijkstra

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

marque à l'étape précédente

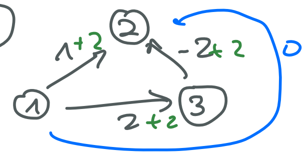
↓

min {∞, 1 + 1} → d(3,5)

↓

marque de 3 à l'étape précédente

②

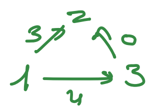


	1	2	3
λ	0	∞	∞
λ		1	2
λ			2

PCCH de 1 à 2 de longueur 1

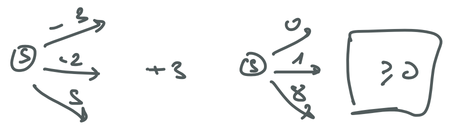
③ Ne marche pas

Contre-exemple.



RQ : marcherait si tous les chemins de s à t avaient le m nb d'arcs

④ OK

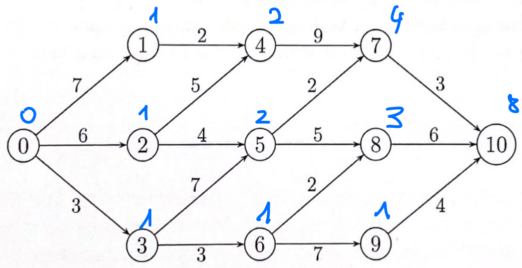


OK

(si pas de circuits 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.



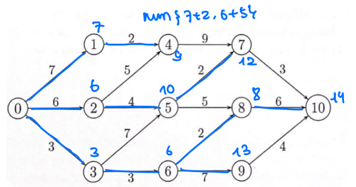
Q 38.1 En utilisant l'algorithme de Bellman, déterminer le tracé dont le coût total de construction est minimum et celui dont le coût total est maximum.

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

Sans circuit → algorithme de Bellman. (ou avec longueur ≤ 0)

Ordre topologique : 0 → 1 2 3 4 5 6 7 8 9 → 10

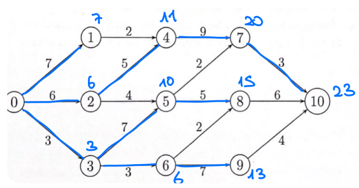
$\lambda(0) = 0$
$\lambda(j) = \min\{\lambda(i) + d(i,j) : i$ prédécesseur de $j\}$



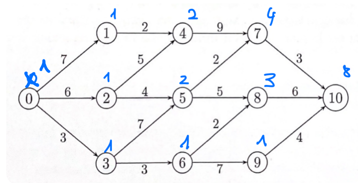
$(0,3,6,8,10)$ de longueur 14.

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

$(0,2,4,7,10)$



Nb de chemins $\begin{cases} \lambda(0) = 1 \\ \lambda(j) = \sum_{i \text{ pred de } j} \lambda(i) \end{cases}$ partition des chemins de 0 à j en fct du dernier sommet avant j



Dans l'ordre : – arcs de poids 1 : parcours en largeur BFS

– graphe sans circuit : Bellman X

– arcs de poids positif : Dijkstra → $i \xrightarrow{3} j$

on regarde les voisins

– Et sinon : Bellman-Ford

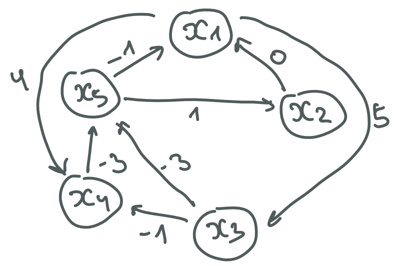
Graphe non orienté

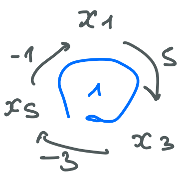
Exercice 43 (Système de différences)

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

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

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

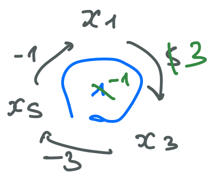




$$\begin{aligned} x_3 - x_1 &\leq 5 \\ x_5 - x_3 &\leq -3 \quad + \\ x_1 - x_5 &\leq -1 \quad + \\ \hline 0 &\leq 1 \end{aligned}$$

$0 \leq 1$

Si circuit absorbant



$$\begin{aligned} x_3 - x_1 &\leq \cancel{5}\, 3 \\ x_5 - x_3 &\leq -3 \quad + \\ x_1 - x_5 &\leq -1 \quad + \\ \hline 0 &\leq \cancel{1} \end{aligned}$$

$0 \leq -1$ faux.

Si le graphe a un circuit absorbant, alors le syst n'a pas de sol°.

Soit $c = (x_{i1}, x_{i2}, \dots, x_{ik}, x_{i1})$ un circuit absorbant (circuit de longueur négatif)

Cela correspond aux inégalités

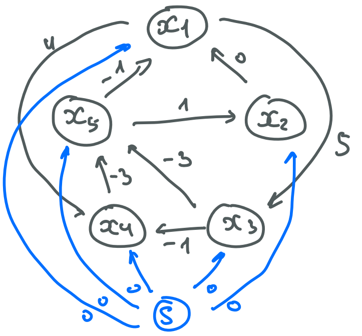
$$\begin{aligned} x_{i2} - x_{i2} &\leq \lambda_1 \\ x_{i3} - x_{i2} &\leq \lambda_2 \quad + \\ &\vdots \\ x_{i1} - x_{ik} &\leq \lambda_k \\ \hline 0 &\leq \sum \lambda_i \end{aligned}$$

en faisant la Σ de ces k inégalités, on obtient :

$\lambda_1 + \lambda_2 \dots + \lambda_k$ est la longueur de circuit (absorbant), donc $\sum_{i=1}^{k} \lambda_i < 0$

On obtient $0 < 0$, contradiction.

On mq si le graphe n'a pas de circuit absorbant, alors le système a une solution.



· Si le graphe contient un circuit absorbant, le syst n'a pas de solution

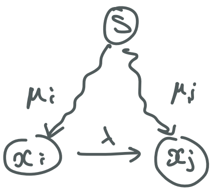
Dans le graphe on ajoute une source s et des arcs (s, x_i) de longueur 0

· Si le graphe ne contient pas de circuit absorbant, soit μ_i la valeur d'un PCCH de s à μ_i

On choisit $x_i = \mu_i$

$x_j - x_i \leq \lambda \Rightarrow x_i \xrightarrow{\lambda} x_j$

L'arc (x_i, x_j) de valeur λ entraîne $\mu_j \leq \mu_i + \lambda$.



$\mu_j \leq \mu_i + \lambda$

Donc $x_j \leq x_i + \lambda$

$x_j - x_i \leq \lambda$.

les x_i sont solutions du syst.

On utilise l'algo de Ford-Bellman, pour déterminer si le graphe a un circuit absorbant et sinon les valeurs de PCCH à partir de s.

	s	x_1	x_2	x_3	x_4	x_5
λ_0	0	∞	∞	∞	∞	∞
p	–	–	–	–	–	–
λ_1	0	0	0	0	0	0
p	–	s	s	s	s	s
λ_2	0	-1	0	0	-1	-3
p	–	x_5	s	s	x_3	x_3
λ_3	0	-4	-2	0	-1	-4
p	–	x_5	x_5	s	x_3	x_4
λ_4	0	-5	-3	0	-1	-4
p	–	x_5	x_5	s	x_3	x_4
λ_5	0	-5	-3	0	-1	-4
p	–	x_5	x_5	s	x_3	x_4

$\lambda_k(x_i)$: valeur d'un PCCH de s à x_i comportant au plus k arcs

$$\underbrace{\lambda_{k+1}(x_j)}_{\text{ligne } k+1} = \min\Big\{ \underbrace{\lambda_k(x_j)}_{x_j \text{ ligne } k}, \min\{ \underbrace{\lambda_k(x_i)}_{\text{ligne } k} + \ell(x_i, x_j) : i \text{ pred } j \}\Big\}$$

$$\lambda_2(x_1) = \min\{0, \underbrace{0+0}_{\text{depuis } s}, \underbrace{0-1}_{\text{" } x_5}, \underbrace{0+0}_{\text{" } x_2}\} = -1$$

seules les marques de x_1 et x_2 ont changé

seuls les marques des successeurs de x_1 et x_2 sont susceptibles de bouger à l'étape d'après.

Les 2 lignes λ_4 et λ_5 sont les m̃ l'algorithme est terminé.

$s \to x_3 \to x_4 \to x_5 \to x_1$, $x_5 \to x_2$

Si pas de circuit absorbant, alors $\exists$ PCCH de s à j qui a au plus $n-1$ arcs

$s \to \to \to \ldots \to k$

$\lambda_n(j) \neq \lambda_{n-1}(j)$ ($<$) contient un circuit absorbant.

$x_1 = -5 \quad x_2 = -3 \quad x_3 = 0 \quad x_4 = -1 \quad x_5 = -5$ est sol° du syst

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 :

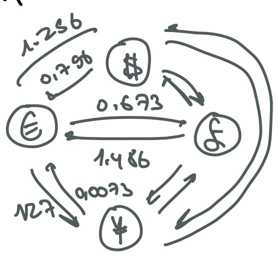
↷	€	\$	£	¥
€	–	1.256	0.673	127
\$	0.796	–	0.536	109
£	1.486	1.867	–	203
¥	0.0073	0.092	0.005	–

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).

Q 39.2 Proposer un algorithme pour résoudre ce problème. Appliquer cet algorithme pour $k = 3$.

Méthode 1



$v(p) = \prod_{v \in p}^{\text{chemin } p} \ell(v)$ → valeur des chemins : produit des taux.

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

c'est k dans l'algo de Ford Bellman

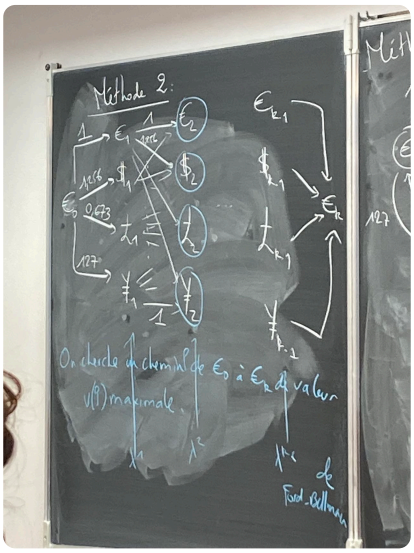
Cela revient à max $\log v(p) = \sum_{v \in p} \log(\ell(v))$

(ou à min $-\log v(p) = \sum_{v \in p} -\log(\ell(v))$)

Algo classique avec $\ell'(v) = -\log \ell(v)$

ou : on applique directement l'algo mais en l'adaptant

(–. max et produit)



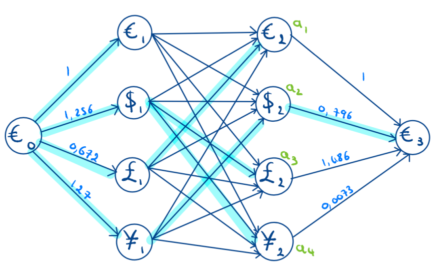
Algo Bellman $v(€_2) = \max\{1 \times 1,\ 1{,}25 \times 0{,}79,\ 0{,}67 \times 1{,}48,\ 127 \cdot 0{,}007\} = a_1$

$v(\$_2) = \max\{1 \times 1{,}256;\ 1{,}256 \times 1;\ 0{,}673 \times 1{,}867;\ 127 \times 0{,}092\} = a_2$

?? $v(£_2) = \max\{1 \times 0{,}673;\ 1{,}256 \times 0{,}536;\ 0{,}673 \times 1;\ 127 \times 0{,}005\} = a_3$

$v(¥_2) = \max\{1 \times 127;\ 1{,}256 \times 109;\ 0{,}673 \times 203;\ 127 \times 1\} = a_4$

$a_1 = 1{,}000078$ $a_2 = 11{,}684$ $a_3 = 0{,}673716$ $a_4 = 136{,}904$.



$v(€_3) = \max\{a_1 \times 1,\ a_2 \times 0{,}796,\ a_3 \times 1{,}486,\ a_4 \times 0{,}0073\}$

Seq optimale € → ¥ → \$ → €

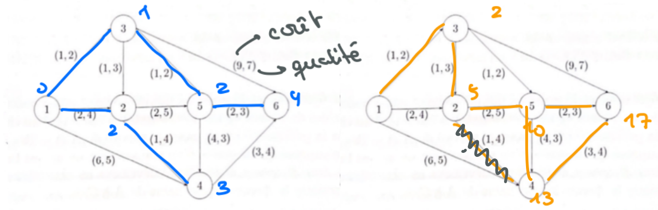
$\max(v(€_3)) = \prod v(e_i)$

$\max(\log v(€_3)) = \max \sum_i \log v(e_i)$

$-\min(-\log v(€_3)) = -\min \sum_i (-\log v(e_i))$

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.

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.

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.

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 $\lambda = \lambda^*$.

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.

(1) Non $(1,3,2,5,4,6)$ pas d'arc en arrière

(2) pas circuit → Bellman $(1,3,5,6)$ coût 4, qualité 7

(3) 1er graphe
$(1,3,2,5,4,6)$ de qualité 17 coût 11
rapport $17/11$.

$\lambda(2) = \min\{\lambda(1) + \ell(1,2), \lambda(3) + \ell(3,2)\}$
$\{0 + 2 \;;\; 1 + 1\}$

2ème graphe
$(1,3,5,6)$ est meilleur que $(1,3,2,5,4,6)$.

$(1,3,2,4,6)$ coût 6 qualité 13
rapport $13/6 > 7/4$

$\max \dfrac{\sum q_v}{\sum_{v \in P} c_v}$

$w_v^\lambda = q_v - \lambda c_v$

Idée : si on arrive à trouver "le bon λ", alors trouver le meilleur chemin selon w_v^λ donne un chemin de $\frac{\text{qualité}}{\text{coût}}$ opt.

$w^\lambda(p) = q(p) - \lambda c(p)$

$w^\lambda(p) \geq 0 \iff q(p) \geq \lambda c(p)$

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

(5) On note p^λ un chemin maximisant $w^\lambda(p) = \sum_{v \in P} w_v^\lambda$

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

Si $w^\lambda(p^\lambda) > 0$ alors $q(p^\lambda) - \lambda c(p^\lambda) > 0$ donc $\dfrac{q(p^\lambda)}{c(p^\lambda)} > \lambda$ et ainsi $\lambda^* > \lambda$

Si $\lambda^* > \lambda$. soit p^* avec $\dfrac{q(p^*)}{c(p^*)} = \lambda^* > \lambda$
$w^\lambda(p^*) = q(p^*) - \lambda c(p^*) > 0$
Ainsi, $w^\lambda(p^\lambda) \geq w^\lambda(p^*) > 0$

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

$w^\lambda(p^\lambda) < 0 \iff \forall p,\ w^\lambda(p) < 0$ $\quad \forall p$ chemin de 1 à 6.
$\iff \forall p,\ q(p) - \lambda c(p) < 0$
$\iff \forall p,\ q(p)/c(p) < \lambda$
$\iff \lambda^* < \lambda$

Ainsi $\lambda = \lambda^* \iff w^\lambda(p^\lambda) = 0$ (et alors p^λ est de $\frac{\text{qualité}}{\text{coût}}$ optimale λ^*)

6. Idée de l'algo
- Trouver M_{min} et M_{max} tq $M_{min} \leq \lambda^* \leq M_{max}$.
- On procède par dichotomie

$\lambda = \dfrac{M_{min} + M_{max}}{2}$ $\quad p^\lambda$ si $w^\lambda(p^\lambda)$ $\begin{cases} = 0 : \text{terminé} \\ > 0 : \lambda < \lambda^* \to M_{min} \leftarrow \dfrac{M_{min} + M_{max}}{2} \\ < 0 : \lambda > \lambda^* \to M_{max} \leftarrow \dfrac{M_{min} + M_{max}}{2} \end{cases}$

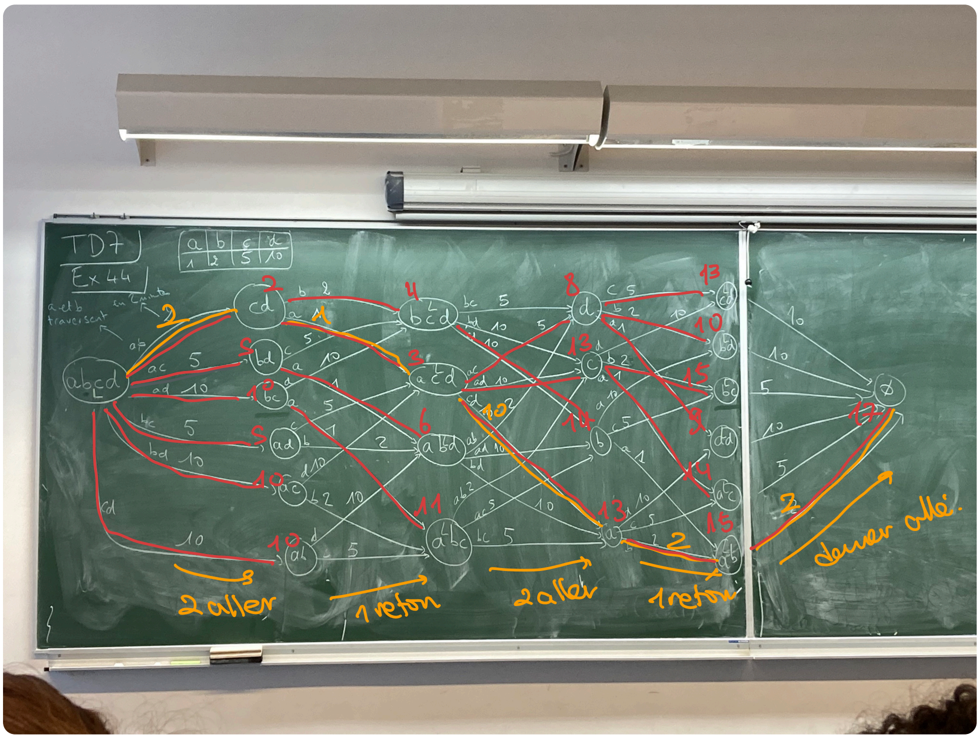
7 Programmation dynamique

Exercice 44 (Franchissement d'un pont)

On considère quatre individus a, b, c, d qui rentrent de randonnée à la nuit tombée. Ils rencontrent un pont suspendu enjambant une rivière. Un panneau indique qu'au maximum deux personnes simultanément peuvent emprunter le pont pour des raisons de sécurité. De plus, un individu ou une paire d'individus doit disposer d'une lampe de poche pour effectuer une traversée. L'individu a met 1 minute pour traverser, b met 2 minutes, c met 5 minutes et d met 10 minutes. Si deux membres du groupe traversent le pont ensemble, ils marchent à la vitesse du membre le plus lent (par exemple, a et c mettent 5 minutes pour traverser ensemble). Le groupe ne disposant que d'une seule lampe de poche, une personne se trouvant après le pont doit retourner au point de départ pour rapporter la lampe après chaque franchissement. On veut déterminer par programmation dynamique l'ordre dans lequel le groupe doit traverser le pont pour minimiser le temps total de franchissement. Pour simplifier la résolution, on ne considérera que des solutions de cinq traversées (allers ou retours) maximum, où les allers se font par deux et les retours seul.

Q 44.1 On définit un état comme la liste des personnes se trouvant avant le pont à l'issue d'une traversée. Représenter le graphe d'états correspondant à ce problème.

Q 44.2 Déterminer le chemin optimal dans le graphe d'états.



État : ens des personnes avant le pont + posi° de la lampe

On cherche un plus court chemin de abcd L à ∅

Graphe sans circuit on peut utiliser l'algo de Bellman

Traversée optimale en 17 minutes

$\xrightarrow{ab}$
$\xleftarrow{a}$
$\xrightarrow{cd}$
$\xleftarrow{b}$
$\xrightarrow{ab}$

Exercice 45 (Organisation des révisions)

Un étudiant de Master M1 dispose de 5 jours pour réviser les UEs LRC, MLBDA et MAPSI (on suppose qu'il connaît parfaitement son cours de MOGPL et qu'il n'a pas besoin de réviser). Il souhaite passer au moins un jour sur chacune des 3 UEs qu'il doit réviser et se demande s'il doit consacrer 1, 2, ou 3 jours à préparer chaque examen. Maîtrisant parfaitement son cours de MOGPL, il décide d'utiliser la programmation dynamique pour trouver la politique de révision qui lui assurera le meilleur total de points (et donc la meilleure moyenne). Il estime pour chaque examen le nombre de points qu'il obtiendra en fonction du nombre de jours qu'il consacrera à l'UE. Ces données sont rassemblées dans le tableau suivant :

nb de jours	LRC	MLBDA	MAPSI
1	6	10	8
2	12	13	14
3	17	16	18

Q 45.1 Dessiner le graphe de programmation dynamique associé à ce problème : pour chaque étape de décision, on représentera les différents états possibles (nombre total de jours consommés) par des sommets, et les arcs associés aux différentes décisions possibles. On précisera également un état initial et un état final. On précisera enfin les valuations des arcs du graphe.

Q 45.2 En utilisant un algorithme du cours (que l'on mentionnera) déterminer le chemin optimal de l'état initial à l'état final dans ce graphe de manière à résoudre le problème posé. Quel est alors la stratégie de révision optimale pour l'étudiant ?

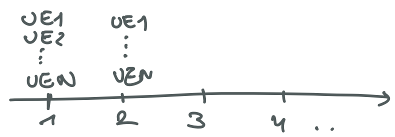
Nb de points max : après 5 jours de révision, sur les 3 UE.

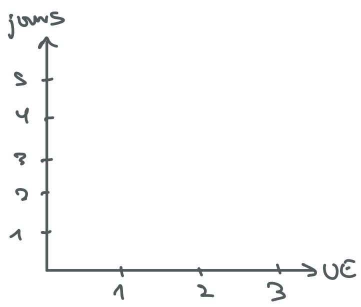
Graphe

UE1 1/2\3

UE2 1/2/3\ 1/2/3\ 1/2/3\

3^N possibilités





UEN : 1 8 +27 =35

2 14 +22 =36

3 18 +16 =34

M = 5 jours.

Gain max total sur les N-1 1° UE

Si on a révisé 4 jours 27

3 jours 22

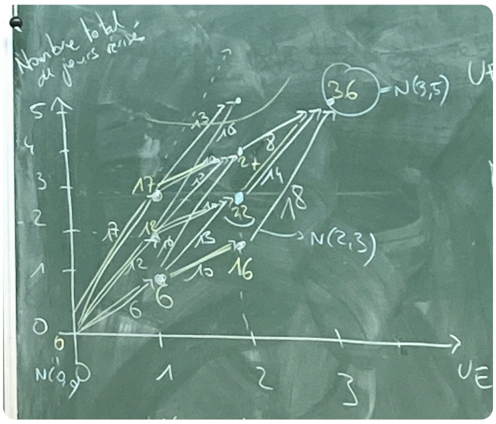
2 jours 16

N(i,j) : meilleur total de pt que l'on peut obtenir en ayant révisé au total j jours sur les i 1° UE.

Pik : pt pour l'UE i si on la révise k jours

On doit calculer un + long chemin du sommet (0,0) au sommet (3,5)

le graphe est sans circuit : on utilise l'algo de Bellman



Le maximum est 36

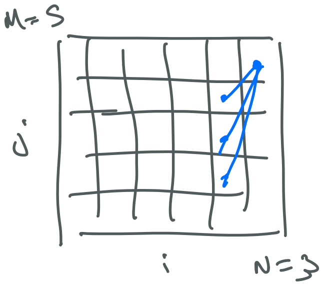
avec 2 jours URC
1 jour MCBDA
2 jours MAPSI

$N(0,j) = 0$

$i > 0$, $N(i,j) = \max \left\{ \begin{array}{l} N(i-1, j-1) + p_{i,1} \ ; \\ N(i-1, j-2) + p_{i,2} \\ N(i-1, j-3) + p_{i,3} \end{array} \right\}$

Initialisation

Pour i de 1 à M
  pour j de 1 à M
    N(i,j) = ...



Algo en O(M.N)

Exercice 46 (Ordonnancement)

Une agence cherche à établir le planning d'une campagne publicitaire pour lancer un nouveau produit. Elle envisage d'utiliser 3 médias : affichage, télévision et annonce de presse. Une conférence de presse est également prévue à la date de lancement officiel du produit. Le premier travail consiste à recenser la liste exhaustive des tâches à accomplir et à estimer les durées de réalisation probables :

Tâches		*Libellé*	*Durée (en semaines)*
a		Préparation du plan de campagne	2
b		Ecriture des textes	4
c	*Presse*	Illustration	4
d		Passation des contrats avec la presse	3
e		Fabrication des plaques de clichés	2
f		Envoi des clichés aux journaux	1
g		Dessin de l'affiche	8
h	*Affichage*	Impression de l'affiche	1
i		Passation du contrat d'affichage	3
j		Distribution des affiches	1
k		Ecriture des scripts	3
l		Etablissement du contrat de tournage	4
m	*Télévision*	Tournage de film	6
n		Passation du contrat avec la T.V.	3
o		Test et envoi du film	1
p		Préparation de la conférence finale	4

Le deuxième travail consiste à établir les contraintes de succession entre les différentes tâches. L'agence se donne deux semaines pour établir le plan détaillé de la campagne (qui précède toute action), après quoi ses représentants commencent à négocier les 4 contrats. Pendant ces négociations, les artistes, dessinateurs et rédacteurs avanceront le plus possible.

Les illustrations destinées aux journaux sont conçues dans le même temps que la rédaction du texte les accompagnant. On ne peut fabriquer les plaques que lorsque le texte et les illustrations sont réalisés. Les plaques sont envoyées aux journaux une fois le contrat de publicité signé.

L'affiche est entre-temps dessinée puis imprimée. La distribution des affiches ne débute qu'après la signature du contrat avec la compagnie d'affichage.

Le texte du film télévisé peut être effectué en même temps que la négociation du contrat de tournage. Quand le film est terminé, il faut l'envoyer à la société de télévision, opération qui ne se fait qu'à l'issue de la signature du contrat.

Ce n'est que lorsque toutes ces opérations sont terminées que l'on peut organiser la conférence de presse qui clora le projet.

On cherche à établir un calendrier précisant la date de début de chaque tâche afin de terminer le projet dans un minimum de temps.

Q 46.1 Soit *fin* une tâche virtuelle qui succède à toute tâche sans successeur. Après avoir défini le dictionnaire des précédents, tracer le graphe *potentiel-tâches* $G = (X, U)$ où :

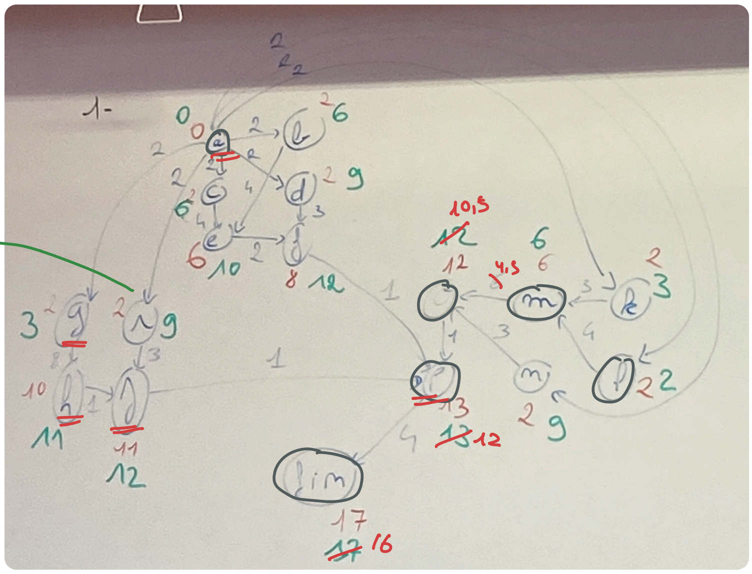
$$\left\{ \begin{array}{c} X = \{a, b, \ldots, p, \mathit{fin}\}, \\ (x, y) \in U \text{ ssi } y \text{ ne peut commencer que lorsque } x \text{ est finie} \end{array} \right.$$

De plus, on affecte à chaque arc (x, y) la valeur $d(x)$ correspondant à la durée de la tâche x.

dictionnaire des précédents

a	∅
b	a
c	a
d	a
e	b, c
f	d, e
g	a
h	g
i	a
j	h, i
k	a
l	a
m	k, l
n	a
o	m, n
p	a…o ou simplement f, j, o
fin	a…p ou simplement p

on peut demander : quand on veut, entre 2 et 9



taches critiques
a, l, m, o, p, fin.

Q 46.2 Déterminer la date de début au plus tôt de m. De façon générale, expliquer à quoi correspond dans G la date de début au plus tôt d'une tâche $x \in X$.

Q 46.3 Montrer que G est sans circuit. Proposer un algorithme adapté pour déterminer la durée minimum de réalisation du projet et l'appliquer au graphe G.

Q 46.4 Déterminer la date au plus tard à laquelle peut commencer e sans que cela ne remette en cause la date de fin au plus tôt du projet. Proposer un algorithme général de calcul des dates au plus tard des différentes tâches du graphe.

Q 46.5 Pour chaque tâche, la différence entre les dates de début au plus tard et au plus tôt est appelée la *marge totale* de la tâche. Les tâches de marges nulles sont appelées *tâches critiques*. Déterminer les tâches critiques du projet.

Q 46.6 Le texte et l'illustration de la campagne de presse sont réalisés par la même personne et ne peuvent avoir lieu en même temps. Cette contrainte modifie-t-elle la durée du projet ?

Q 46.7 On désire lancer la conférence de presse dans 16 semaines. Pour ce faire, on peut réduire le temps de réalisation de certaines tâches (p exceptée) en exploitant des heures supplémentaires. Pour une tâche, les heures supplémentaires ne peuvent excéder 25% de la durée initiale prévue. Indiquer sur quelle tâche affecter ces heures.

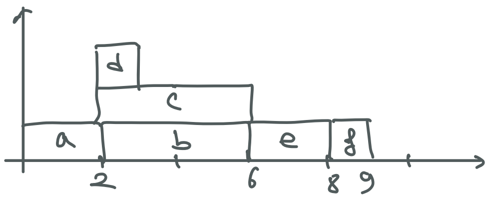
2. $\text{date}(m) = \max\{\text{date}(k)+3, \text{date}(l)+4\}$
$= \max\{5, 6\}$
$= 6$

De façon generale, date(x) = max { chemins de a vers x }

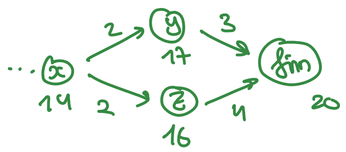
3. La durée minimale du projet est de 17 semaines.

Par l'algo de Bellman adapté au cas de + long chemin.

(graphe sans circuit : a, b, …, p, fin est un ordre topologique)



Pour finir à 17, on peut commencer e au plus tard à



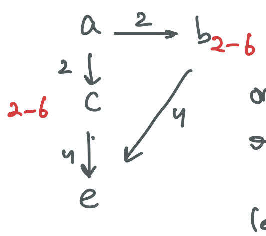
date de début au + tard de x :
durée_projet − longueur du + long chemin de x à fin

→ Dans un ordre topologique inverse

$$d_{\text{au + tard}}(x) = \min \{ d_{\text{au + tard}}(y) - d(x) \ : \ y \in \underbrace{\Gamma^+(x)}_{\text{successeurs}} \}$$

ou : on calcule les + longs chemins de chq tâche x à fin et on fait la diff.
(avec une seule fois Bellman)

6. b et c ne peuvent pas être faites en parallèle.

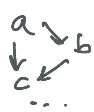


on a de la marge
on peut faire commencer b à 2
et c à 6
(et alors e à 10)

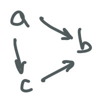
→ pas de retard

. De manière générale : 2 tâches b et c ne peuvent pas être faits en parallèle

→ Soit b avant c



→ Soit c avant b



on doit traiter les 2 cas.

17 → 16 semaines.

on peut, avec les hrs supplémentaires (HS), réduire la durée d'une tâche de 25%.

Oui en passant la durée de m à 4 S. ⟶ tâches critiques a, l, m, o, p, fin
↓
a, g, h, j, p, fin.

Exercice 47 (Distance d'édition de deux chaînes)

On s'intéresse à la meilleure façon de remplacer une chaîne de caractères alphabétiques par une autre avec un système de traitement de texte (tel que Word par exemple). On supposera pour cela que l'on dispose des opérations élémentaires suivantes :

COPIER : supprime le caractère le plus à gauche de la chaîne-source et place ce caractère à droite de la chaîne-cible ;

REMPLACER : supprime le caractère le plus à gauche de la chaîne-source et place un caractère différent à droite de la chaîne-cible ;

SUPPRIMER : supprime le caractère le plus à gauche de la chaîne-source (modifie seulement la chaîne-source) ;

INSÉRER : place un caractère non emprunté à la chaîne-source en position à droite de la chaîne-cible (modifie seulement la chaîne-cible).

On supposera que chaque opération a un *coût* qui reflète le temps moyen mis par un opérateur expérimenté pour effectuer cette opération. On supposera ici que les coûts des diverses opérations ci-dessus sont :

$C_C = 2$ (COPIER) $C_R = 5$ (REMPLACER) $C_S = 3$ (SUPPRIMER) $C_I = 4$ (INSÉRER)

Remarque : on vérifie bien $C_R < C_S + C_I$, sans quoi pour réaliser un remplacement on aurait intérêt à effectuer une suppression suivie d'une insertion.

Exemple : pour obtenir IVRY à partir de CLICHY on peut par exemple : SUPPRIMER C, SUPPRIMER L, COPIER I, SUPPRIMER C, INSÉRER V, REMPLACER H par R, COPIER Y. Cela fait un coût total de $3C_S + 2C_C + C_I + C_R = 22$. Une meilleure solution consiste à remplacer les deux opérations SUPPRIMER C et INSÉRER V par l'unique opération REMPLACER C par V, ce qui permet d'obtenir un coût total de 20 au lieu de 22.

Etant données deux chaînes x (chaîne-source) et y (chaîne-cible), on appelle *distance d'édition* le coût de la séquence d'opérations la plus économique pour transformer x en y.

Q 47.1 On note $D(i, j)$ la distance d'édition entre la sous-chaîne constituée par les i derniers caractères de x et les j derniers caractères de y. Établir la relation de récurrence générale permettant de déterminer les valeurs $D(i, j)$ pour tout i compris entre 0 et n (nombre de caractères de x) et pour tout j compris entre 0 et m (nombre de caractères de y). Expliquer comment initialiser cette récurrence.

Q 47.2 Quelle information faut-il conserver, et sous quelle forme, à chaque étape de la résolution de la récurrence, pour obtenir non seulement la valeur optimale du problème ($D(n, m)$) mais également la suite d'opérations correspondante.

Q 47.3 Trouver la distance d'édition pour x = "AGORA" et y = "GARE" et expliciter la suite des opérations permettant de transformer x en y.

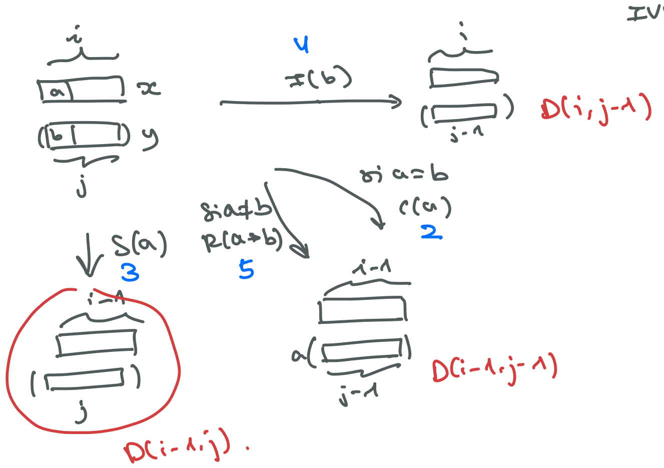
CLICHY (IVRY) $\xrightarrow[3]{S(C)}$ LICHY (IVRY) $\xrightarrow[3]{S(L)}$ ICHY (IVRY) $\xrightarrow[2]{C(I)}$ CHY I(VRY) $\xrightarrow[3]{S(C)}$ HY I(VRY) $\xrightarrow[4]{I(V)}$ HY IV(RY) $\xrightarrow[5]{R(H \to R)}$ Y IVR(Y) $\xrightarrow[2]{C(Y)}$ ∅ IVRY

R(C→V) 5

Copier : 2
remplacer : 5
Supprimer : 3
Insérer : 4

Coût 20
Coût 22

(1) D(i,j)



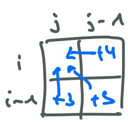
Si a≠b. $D(i,j) = \min\{D(i-1,j)+3,\ D(i-1,j-1)+5,\ D(i,j-1)+4\}$

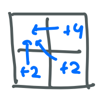
Si a=b $D(i,j) = \min\{D(i-1,j)+3\ ;\ D(i-1,j-1)+2,\ D(i,j-1)+4\}$

• i=0 (x est vide) $D(0,j) = 4j$

• j=0 (y est vide) $D(i,0) = 3i$

$\to O(|x||y|)$

Cas a≠b 

Cas a=b 

i \ j	G 4	A 3	R 2	E 1	∅ 0
A 5	17	15	16	17	15
G 4	14	15	13	14	12
O 3	16	12	10	11	9
R 2	15	11	7	8	6
A 1	14	10	9	5	3
∅ 0	16	12	8	4	0

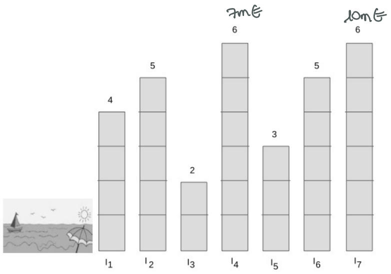
14
14 14 16
11 11 13

∅ ARE

S(A) C(G) R(O→A) CCR R(A→E

Exercice 48 (Vue sur la mer [1])

La construction sauvage d'immeubles sur la côte d'azur a débouché sur la situation de la figure ci-dessous dans la ville de Saint-Trépoz.



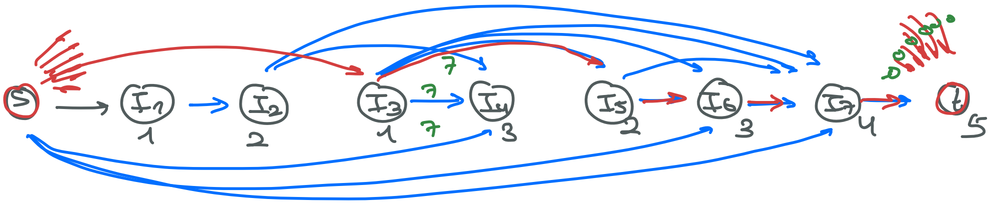
Cette figure représente $n = 7$ immeubles $I_1, \ldots, I_7$, chacun immeuble I_i ayant un nombre d'étages $t(i)$. Devant l'immeuble 1 se trouve l'étendue infinie de la mer méditerrannée. Comme on peut le constater, des immeubles bouchent intégralement la vue sur la mer de certains immeubles situés en arrière (par exemple l'immeuble I_2 bouche la vue de l'immeuble I_3). On dira qu'un immeuble I_i bouche la vue sur la mer d'un immeuble I_j en arrière de I_i ($j > i$) si (et seulement si) $t(i) \geq t(j)$.

Elu(e) à la mairie de Saint-Trépoz, vous mettez en place une nouvelle politique radicale : votre but est que chaque immeuble ait vue sur la mer. Pour cela, vous allez conserver certains immeubles, et détruire les autres ! Vous souhaitez bien entendu maximiser le nombre d'immeubles qui seront conservés. Ceux-ci doivent donc tous avoir vu sur la mer, aucun immeuble conservé ne doit boucher intégralement la vue d'un autre.

NB : Un immeuble est soit conservé intégralement, soit détruit complètement - on ne peut pas détruire juste quelques étages d'un immeuble.

Proposez un algorithme de programmation dynamique de complexité $O(n^2)$ pour trouver le nombre maximum d'immeubles que l'on peut conserver. Vous spécifierez bien la signification des éléments calculés et la relation de récurrence, et justifierez brièvement la complexité de l'algorithme. Appliquez votre algorithme sur l'exemple.

Solution optimale $\{3,5,6,7\}$ → 4 immeubles conservés



chemin) ens d'immeubles (réalisable) de valeur maximale

$I_i \to I_j$: $i<j$ et $t(i) < t(j)$

ok. I_i ne bouche pas la vue à I_j

$O(n^2)$ test pour chq paire (i,j) $O(n^2)$.

Chemin ⇔ ens d'immeubles qu'on peut conserver

+ long chemin de s à I_j : + grande séq d'immeubles (réalisable) dont le dernier est I_j

+ long ———— t : séquence optimale

Graphe sans circuit → algo de Bellman $O(m+n) = O(n^2)$

n immeubles → $m = O(n^2)$ arcs dans le graphe.

$\underbrace{\lambda(j)}$ = max $\{1 + \lambda(i) : i<j$ et $t(i) < t(j)\}$

si le dernier immeuble conservé avant I_j est I_i

n marges à calculer $O(n^2)$

8 Problèmes de flot maximum

Exercice 49 (Problème d'import-export)

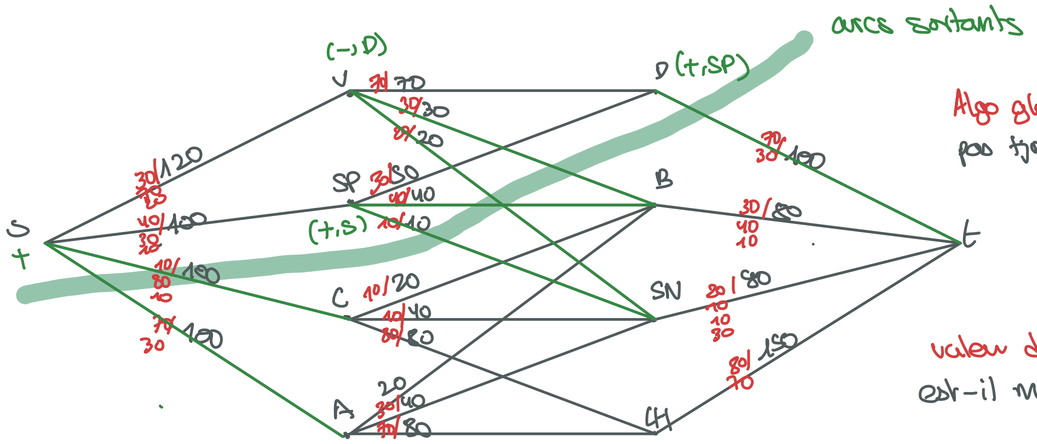
Une société d'import-export dispose, dans les ports de Veracruz, Sao Paulo, Conakry et Abidjan, de stocks de café de respectivement 120t, 100t, 100t et 100t, pour lesquels elle a reçu des commandes d'importateurs de Dunkerque (100t), Bordeaux (80t), Saint-Nazaire (90t) et Le Havre (150t). Divers bateaux se rendent des ports étrangers considérés vers les ports français de destination. Les tonnages sont donnés par le tableau suivant :

	Dunkerque	Bordeaux	Saint-Nazaire	Le Havre
Veracruz	70	30	20	
Sao Paulo	50	40	10	
Conakry		20	40	80
Abidjan		20	40	80

Q 49.1 Modéliser ce problème sous la forme de la recherche d'un flot maximum dans un réseau de transport.

Q 49.2 Déterminer les diverses cargaisons de façon à satisfaire au mieux les demandes, les commandes destinées à Bordeaux et au Havre étant prioritaires.

1.



Algo glouton
pas tjr le flot max

valeur de flot = 400
est-il max ?

TH. Flot max Coupe min : Si coupe = 400, alors c'est bon ?

Algo de Marquage

(S, SP, V, D), (A, B, SN, LH, t, C)

→ 100 + 30 + 20 + 40 + 10 + 100 + 100 = 400. donc bien flot max

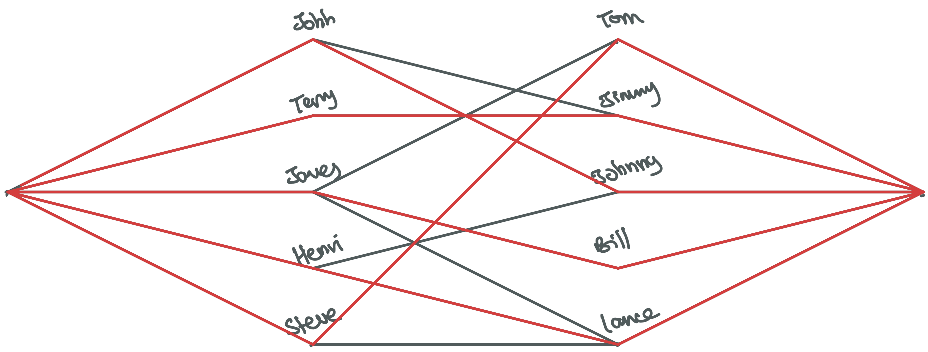
Exercice 50 (Bataille d'Angleterre)

Lors de la seconde guerre mondiale, la bataille d'Angleterre se déroula surtout dans le ciel. Les biplaces anglais comportait un pilote et un mitrailleur. Du fait des besoins de coordination entre les deux équipiers, le pilote et le mitrailleur devaient avoir réalisé des sorties d'entraînement ensemble pour pouvoir ensuite sortir ensemble en conditions réelles. Le tableau ci-dessous, comportant les pilotes en ligne et les mitrailleurs en colonne, indique les équipiers potentiels ayant déjà réalisé des sorties d'entraînement ensemble. Les Anglais visaient bien évidemment à composer des équipes permettant d'avoir le plus possible d'avions simultanément dans le ciel.

	Tom	Jimmy	Johnny	Bill	Lance
John		X	X		
Terry		X			
James	X			X	X
Henry			X		X
Steve	X				X

Q 50.1 Modéliser ce problème sous la forme d'un problème de flot maximum dans un réseau de transport.

Q 50.2 En première approche, on a formé les équipes (John,Johnny), (Terry,Jimmy), (James,Bill) et (Steve,Lance). Proposer une meilleure solution.



Exercice 51 (Stable maximum)

Un individu souhaite organiser une soirée regroupant plusieurs de ses amis. Cependant, afin de préserver une ambiance conviviale, il sait qu'il vaut mieux éviter de mettre en présence certains de ceux-ci. Les incompatibilités de caractères sont représentées sur le graphe biparti G de la figure 1, dont les sommets figurent les différents amis et les arêtes figurent les incompatibilités. La détermination d'un groupe d'amis le plus large possible à inviter correspond à la recherche d'un ensemble stable de cardinalité maximale sur G. On rappelle qu'un ensemble stable d'un graphe non orienté est un sous-ensemble S de sommets tel qu'aucune arête n'a ses deux extrêmités dans S.

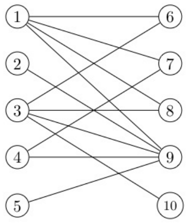


FIGURE 1 – Graphe des incompatibilités de caractères.

Les deux questions qui suivent visent à montrer que la détermination d'un ensemble stable de cardinalité maximale sur G peut se ramener à la résolution d'un problème de flot maximum.

Q 51.1 Montrer que le complément dans X (ensemble des sommets) de toute couverture est un ensemble stable, et réciproquement que le complément de tout ensemble stable est une couverture. En déduire que rechercher un stable de cardinalité maximale est équivalent à rechercher une couverture de cardinalité minimale. On rappelle qu'une couverture est un sous-ensemble de sommets tel que toute arête du graphe a au moins une de ses extrêmités dans le sous-ensemble.

Q 51.2 Soient X_1 et X_2 les deux sous-ensembles stables de sommets en lesquels X se partitionne. Construire un graphe orienté G' à partir de G, muni d'une source s et d'un puits t, puis déterminer des capacités sur les arcs de G' de façon à ce que la valeur de la coupe minimum séparant s et t dans G' soit égale à la valeur d'une couverture minimum dans G. Pour le prouver, montrer tout d'abord que si (S, T) est une coupe minimum dans G', obtenue au terme de l'algorithme de Ford et Fulkerson, alors l'ensemble $(X_1 \cap T) \cup (X_2 \cap S)$ est une couverture sur G de même valeur. Montrer ensuite que pour toute couverture C dans G, la valeur de la coupe $(\{s\}\cup(\bar{C}\cap X_1)\cup(C\cap X_2), \{t\}\cup(C\cap X_1)\cup(\bar{C}\cap X_2))$ dans G' est égale à la cardinalité de C.

Q 51.3 Appliquer l'algorithme de Ford et Fulkerson pour déterminer un flot maximum sur le graphe construit dans la question précédente. On expliquera de façon claire et détaillée les étapes de successives de la détermination de ce flot maximum. Expliquer ensuite en détail comment on peut déduire de ce calcul un ensemble stable de cardinalité maximale sur G. On indiquera précisément la liste des sommets composant cet ensemble stable.

Q 51.4 Supposons maintenant que chacun des amis ne présente pas la même utilité pour leur hôte. On souhaite donc prendre en compte cet aspect dans le modèle. Pour ce faire, à chaque sommet i du graphe G on associe un *poids* noté $w(i)$, dont la valeur, pour chaque sommet, est précisée dans le tableau suivant :

i	1	2	3	4	5	6	7	8	9	10
$w(i)$	4	2	3	3	5	2	1	4	5	1

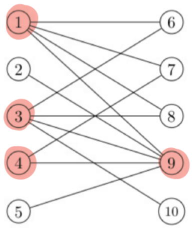
On recherche maintenant un ensemble stable de G *de poids maximal*, le poids d'un sous-ensemble de sommets étant égal à la somme des poids des sommets qui le composent. Montrer pourquoi et comment ce problème peut, lui aussi, se ramener à la résolution d'un problème de flot maximum dans un graphe qu'on précisera, en indiquant clairement les capacités attribuées à chacun des arcs.

Q 51.5 Résoudre le problème de flot maximum sur le graphe proposé à la question 4 et préciser l'ensemble stable de poids maximum obtenu.

Stable max ⇔ Couverture min $\{1,2,3,4,5\}$

Q. $\exists$ couverture de cardinalité <5?

$C = \{1,3,9,4\}$



le complement de C est X/C

①

* Le complement d'un stable est une couverture

Soit S le stable et soit $\bar{S} = X/C$ son complement.

Par l'absurde, on suppose que $\bar{S}$ n'est pas une couverture.

$\Rightarrow \exists$ arête dont aucune extremité ne fait partie de $\bar{S}$.

$\Rightarrow$ ses 2 extremités font partie de S. Contradiction.

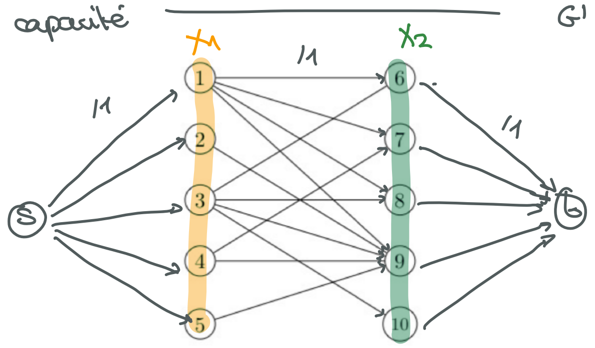
* Le complement d'une couverture est un stable

Soit C la couverture et $\bar{C}$ son complement

Par l'absurde, supp que $\bar{C}$ n'est pas stable

$\Rightarrow \exists$ arête dont les 2 extrémités sont dans $\bar{C}$ et donc C n'est pas une couverture.

La taille d'une couverture minimale dans G.
capacité ______ G'



<u>Obs.</u> La capacité d'une coupe min est finie.

Est-ce que la capacité d'une coupe min dans G' peut être infinie ?

Supposons que $(X_1 \cap T) \cup (X_2 \cap S)$ n'est pas une couverture.
Dans ce cas, il existe une arête $e = (\underset{\in X_1}{i}, \underset{\in X_2}{j})$ tq $i \notin (X_1 \cap T) \cup (X_2 \cap S)$
et $j \notin (X_1 \cap T) \cup (X_2 \cap S)$

Donc forcément, on a $i \notin T$
Comme (S,T) forme une partition de l'ens de sommet, on en déduit que $i \in S$,
Donc le sommet i est marqué.
D'après l'algo de marquage, on déduit que j est forcément marqué, donc $j \in S$.
Or $j \notin X_2 \cap S$ par hypothèse.

Donc contradiction.

$$v(S,T) = \sum_{i \in X_1 \cap T} c(s,i) + \sum_{i \in X_2 \cap S} c(i,t)$$

valeur coupe min

$$= |X_1 \cap T| + |X_2 \cap S|$$

= taille de couverture.

Soit C une couverture dans G.

Mq la coupe $(\underbrace{\{s\} \cup (\bar{C} \cap X_1) \cup (C \cap X_2)}_{S'}, \underbrace{\{t\} \cup (C \cap X_1) \cup (\bar{C} \cap X_2)}_{T'})$ a la m̂ valeur que C, cad qu'elle est égale à $|C|$

$$v(S', T') = \sum_{i \in C \cap X_1} c(s,i) + \cancel{\sum_{\substack{i \in \bar{C} \cap X_1 \\ j \in \bar{C} \cap X_2}} c(i,j)} + \sum_{i \in C \cap X_2} c(i,t)$$

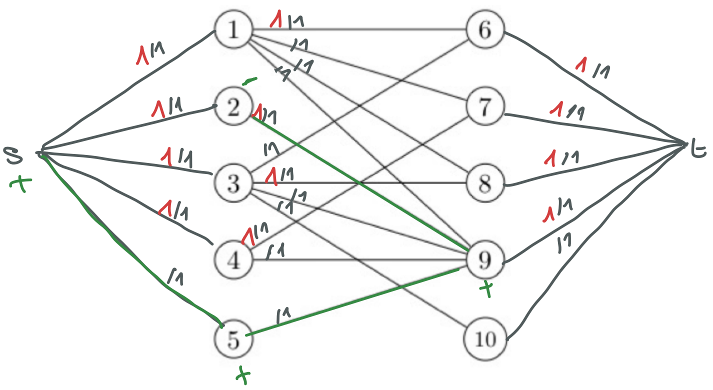
car C couverture.

$$= |C \cap X_1| + |C \cap X_2|$$

$$= |C|$$

Ford Fulkerson

Flot complet   Algo de marquage



$C = (S,T)$ avec $S = \{s, 5, 9, 2\}$

↑

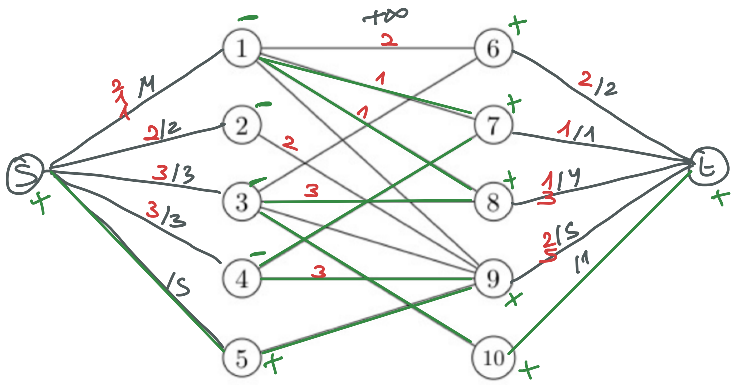
coupe min

$T = \{t, 1, 3, 4, 6, 7, 8, 10\}$

La couverture min est donc
$(X_1 \cap T) \cup (X_2 \cap S) = \{1,3,4\} \cup \{9\}$
$= \{1,3,4,9\}$

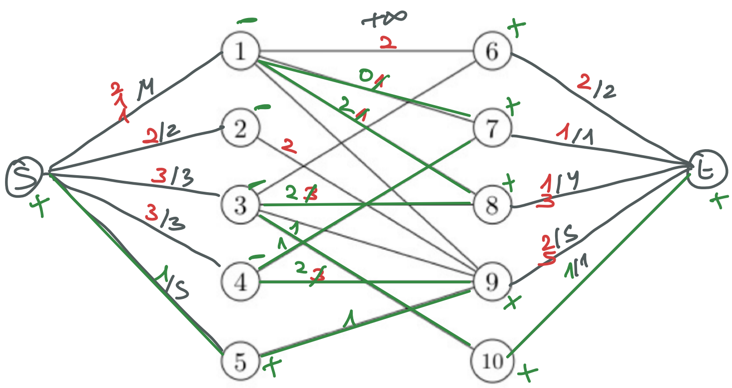
Donc l'ensemble stable de taille max est obtenu en prenant le complémentaire

on obtient donc $\{2,5,6,7,8,10\}$

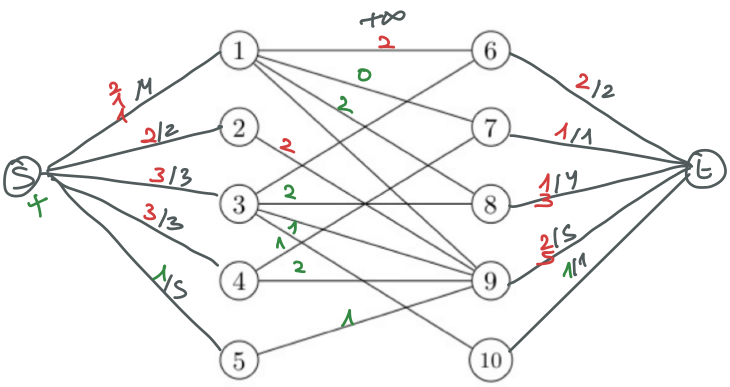


On cherche à avoir flot complet du début à la fin.

On fait algo marquage.



On rajoute 1 pour arc avant
−1 pour arc arrière.



on refait m̂ chose

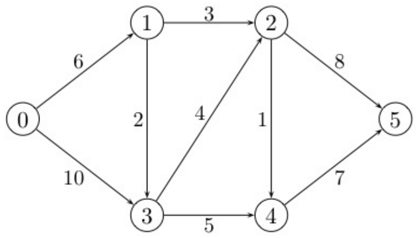
$C = (S,T)$ avec $S = \{s,2,4,5,7,9\}$
$T = \{t,1,3,6,8,10\}$

couverture min $(X_1 \cap T) \cup (X_2 \cap S)$
$= \{1,3\} \cup \{7,9\}$
$= \{1,3,7,9\}$

Stable max : $\{2,4,5,6,8,10\}$

Exercice 52 (Minimisation d'une fonction quadratique)

On considère le graphe orienté suivant dont les arcs sont munis de capacités.



Q 52.1 Déterminer un flot maximum et une coupe minimale séparant 0 de 5. On associe maintenant une variable 0-1 à chaque sommet du graphe en attribuant la valeur 1 à x_0 et la valeur 0 à x_5, et on considère la coupe séparant le sous-ensemble $S = \{i : x_i = 1\}$ de $\bar{S} = \{i : x_i = 0\}$. Montrer que la capacité de cette coupe peut s'exprimer comme une fonction quadratique (i.e., de degré 2) des variables x_i. En déduire que la recherche d'une coupe minimale séparant deux sommets donnés dans un graphe peut se ramener à la minimisation d'une fonction quadratique en variables 0-1. Que peut-on dire des coefficients des termes quadratiques dans la fonction obtenue ?

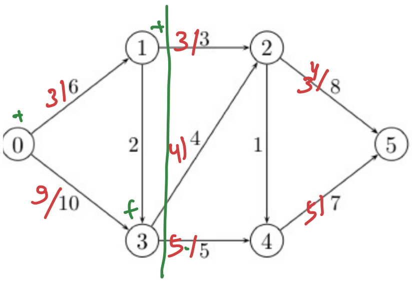
Q 52.2 Soit la fonction quadratique :

$$g(x) = -2x_1 + 15x_2 - 3x_3 + 10x_4 - 4x_1x_2 - 3x_1x_3 - 2x_2x_3 - 9x_2x_4 - 5x_3x_4$$

Montrer que la minimisation de cette fonction peut se ramener à la recherche d'une coupe minimale (donc d'un flot maximum) sur un graphe que l'on construira. En déduire la valeur de $\min_{x \in \{0,1\}^4} \{g(x)\}$.

Q 52.3 Pourrait-on faire la même construction qu'à la question précédente pour la fonction :

$$h(x) = -2x_1 + 15x_2 - 3x_3 + 10x_4 - 4x_1x_2 + 3x_1x_3 - 2x_2x_3 + 9x_2x_4 - 5x_3x_4$$



Coupe min $C = (S,T)$ avec $S = \{0,1,3\}$
$T = \{2,4,5\}$

Soit $C = (S,T)$ la coupe défini par les variables x_i.

$$v(s,t) = \sum_{\substack{(i,j) \in E \\ \text{avec } i \in S \\ j \in T}} c_{ij} = \sum_{(i,j)} c_{ij} \times x_i (1 - x_j)$$

Donc trouver la coupe de valeur min revient à minimiser la fct

$$f(x) = \sum_{(i,j) \in E} c_{ij} x_i (1 - x_j)$$

Le coefficient des termes quadratiques sont négatifs

9 Problèmes d'affectation et de transport

Exercice 54 (Site de rencontres sur internet)

On considère un site de rencontre sur internet, où les membres à la recherche de l'âme-sœur sont invités à "tchater" entre eux dans un premier temps. A l'issue de ces discussions, les membres sont invités à attribuer un score à leurs divers interlocuteurs. Le webmaster doit organiser les rencontres réelles qui auront lieu dans les prochains jours, sur la base de ces informations. Voici les tableaux des scores qui ont été attribués par les garçons aux filles, et par les filles aux garçons :

	Eve	Juliette	Cléopâtre	Yseult	Dalila
Adam	7	6	7	6	8
César	9	8	8	9	7
Roméo	7	6	5	10	8
Samson	4	9	8	4	9
Tristan	5	9	8	8	9

Table 1 – Scores attribués par les garçons aux filles

	Eve	Juliette	Cléopâtre	Yseult	Dalila
Adam	9	9	8	5	4
César	8	5	9	8	5
Roméo	7	9	7	5	7
Samson	10	6	7	6	10
Tristan	6	4	6	7	9

Table 2 – Scores attribués par les filles aux garçons

Q 54.1 On souhaite organiser des rendez-vous entre les garçons et les filles (un garçon par fille et réciproquement) de façon à minimiser la somme totale des regrets des différents protagonistes (le regret d'un protagoniste est la différence entre la note de son interlocuteur préféré et la note de l'interlocuteur qu'il va être amené à rencontrer). Modéliser ce problème comme un problème d'affectation.

Q 54.2 Résoudre à l'aide de l'algorithme hongrois le problème ainsi posé.

Q 54.3 Suite à la forte croissance du nombre des membres sur le site, il ne devient plus possible pour chaque membre de discuter avec tous les membres du sexe opposé. C'est pourquoi le webmaster adopte une nouvelle procédure qui consiste à demander à chaque membre d'indiquer les membres du sexe opposé qu'il connaît. Il s'agit donc d'organiser des rendez-vous entre des personnes qui se connaissent. On suppose que le nombre n de garçons est égal au nombre de filles. En modélisant le problème comme un problème de flot, montrer que si chaque fille connaît exactement $k \geq 1$ garçons et si chaque garçon connaît exactement k filles, alors on peut arranger n rendez-vous entre des gens qui se connaissent.

Regret des garçons

	E	J	C	Y	D
A	1	2	1	2	0
Ce	0	1	1	0	2
R	3	4	5	0	2
S	5	0	1	5	0
T	4	0	1	1	0

Regret des filles

	E	J	C	Y	D
A	1	0	1	3	6
Ce	2	4	0	0	5
R	3	0	2	3	3
S	0	3	2	2	0
T	4	5	3	1	1

Regret total

	E	J	C	Y	D	
A	2	2	2	5	6	← -2
Ce	2	5	1	0	7	
R	6	4	7	3	5	← -3
S	5	3	3	7	0	
T	8	5	4	2	1	← -1
	≥0					

enleve 2 à la valeur de toute solution

-6 à la valeur de toute sol°

non marquées

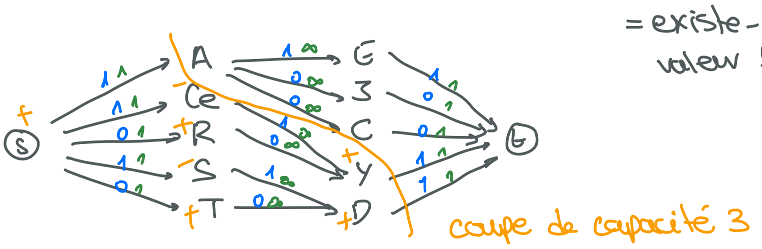
marquées

	E	J	C	Y	D	
A	0	0	0	3	4	
Ce	2	5	1	0	7	-1
R	3	1	4	0	2	-1
S	5	3	3	7	0	-1
T	7	4	3	1	0	-1
				+1	+1	

Toute colonne contient au moins un 0.

Existe-t-il une nouvelle solution de coût 0 ? (nouvelle matrice)

Graphe des "0"



= existe-t-il un flot de valeur 5 dans ce graphe ?

→ Faire apparaître des arcs qui sortent de la coupe

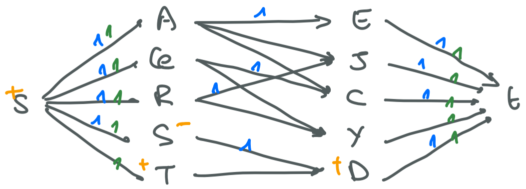
→ D'un sommet-ligne marqué à un sommet-colonne non marqué → ça coût 0

Elément min $\delta = 1$, -1 sur les lignes marqués.
$+1$ sur les colonnes marqués

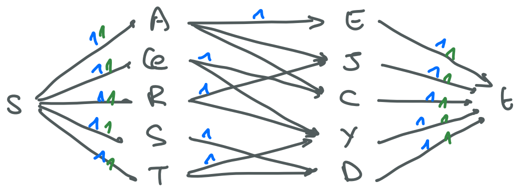
non marqué

	E	J	C	Y	D	
A	0	0	0	4	5	
Ce	1	4	0	0	7	
R	2	0	3	0	2	
S	4	2	2	7	0	−1
T	6	3	2	1	0	−1
					+1	

marqués



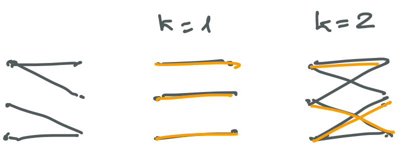
	E	J	C	Y	D
A	0	0	0	4	6
Ce	1	4	0	0	8
R	2	0	3	0	3
S	3	1	1	6	0
T	5	2	1	0	0

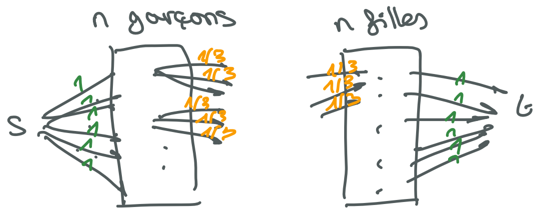


On a le couplage A-E Ce-C R-J S-D T-Y (Solution optimale)

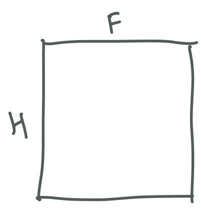
de coût regret 9.

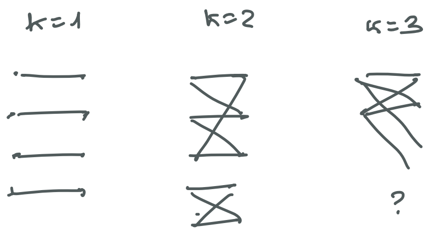
$-6\,(4+2)\,(-2+1) = -9$ coût 0 à la fin





Nb max de rdv (avec au plus un RDV par personne) = valeur flot max





Flot de valeur n en mettant $\frac{1}{k}$ sur chq arc entre 1 garçon et une fille.

les capacités sont entières. donc $\exists$ un flot entier de valeur n.

→ On peut tjr organiser n RDV

Exercice 55 (Gestion d'un projet informatique)

La société Macrosoft SA veut réaliser un projet de logiciel qui consiste en 4 tâches : construction d'une base de données, interface utilisateur, modélisation du problème et algorithmes d'optimisation. Ces 4 tâches doivent être confiées aux 4 meilleurs ingénieurs de la société (I1, I2, I3 et I4). Le tableau suivant donne le temps en jours dont chaque ingénieur a besoin pour accomplir les différentes tâches :

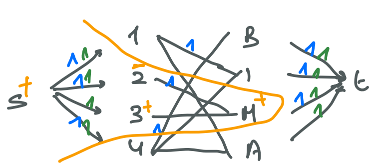
	base de données	interface	modélisation	algorithmes	
I1	8 7	1 0	5 4	5 4 0	-1
I2	5 1	6 2	4 0	11 11 3	-4
I3	6 3	4 1	3 0	9 6 2	-3
I4	3 0	3 0	7 4	7 4 0	-3
				↑ -4	

Q 55.1 Utilisez la méthode Hongroise pour résoudre le problème de minimisation du temps total nécessaire pour réaliser le projet. Donnez l'affectation optimale et le temps total obtenu.

Q 55.2 La solution optimale obtenue dans la question 1 est elle unique ? Sinon, expliquez comment obtenir d'autres solutions de même valeur pour la fonction objectif, puis appliquez le principe proposé sur l'instance de la question 1 et explicitez la ou les solution(s) obtenue(s).

	B	I	M	A	
1	7	0	4	0	
2	1	2	0	3	-1
3	3	1	0	2	-1
4	0	0	4	0	
			+1		

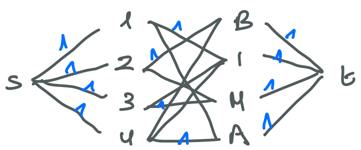
Graphe des 0



Sommets lignes marqués : {I2, I3}
Sommets colonnes marqués : {M}

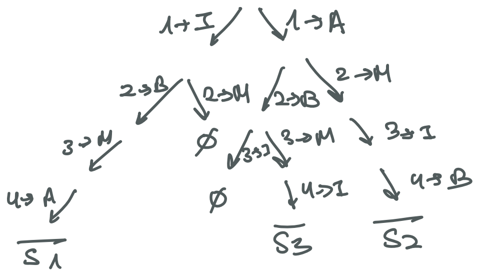
	B	I	M	A
1	7	0	5	0
2	0	1	0	2
3	2	0	0	1
4	0	0	5	0

Graphe des 0



On a 1 → I
2 → B
3 → M
4 → A
de durée 16

ou 1 → A , 2 → M 3 → I 4 → B



On regarde les 0 de la matrice et on fait un arbre de recherche

Exercice 57 (Location de voitures)

Une entreprise de location de voitures cherche à redistribuer *à moindre coût* les voitures dans ses agences de manière à ce que le nombre de voiture dans chaque agence corresponde au mieux à ses besoins. Actuellement, l'entreprise a 10 voitures de trop à Rennes et 12 de trop à Paris. En revanche, ses agences à Orléans, Nice et Bordeaux aimeraient obtenir respectivement 5, 11 et 6 voitures supplémentaires. Les coûts de transport (en Euros) d'une voiture d'une ville à une autre sont donnés ci-dessous :

	Orléans	Nice	Bordeaux
Rennes	50	250	100
Paris	25	200	125

Q 57.1 Montrez que l'on peut modéliser ce problème comme un problème de transport et résolvez-le par l'algorithme primal-dual.

Q 57.2 On suppose maintenant que les agences d'Orléans, de Nice et de Bordeaux souhaitent obtenir respectivement 6, 14 et 7 voitures supplémentaires. Le nombre de voitures souhaitées est donc plus grand que le nombre de voitures disponibles. On suppose de plus qu'une agence peut obtenir moins de voitures que son souhait sans que cela n'engendre de coût supplémentaire. Montrez que l'on peut encore modéliser ce problème comme un problème de transport et résolvez-le par l'algorithme primal-dual.

Indication : Introduire un sommet fictif.

Q 57.3 Il s'avère maintenant que si l'agence d'Orléans obtient moins de voitures qu'elle ne le souhaite, cela engendre un coût de 50 Euros pour chaque voiture manquante. De même, pour les agences de Nice et de Bordeaux, il en coûte respectivement 75 et 100 Euros pour chaque voiture manquante. Modifiez la modélisation donnée en question 2 pour prendre en compte ces changements (on ne demande pas ici de résoudre ensuite le problème).

	Orléans	Nice	Bordeaux	Offre	
Rennes	50	250	100	10	−50
Paris	25	200	125	12	−25
Demande	5	11	6	22	

Demande totale : 22

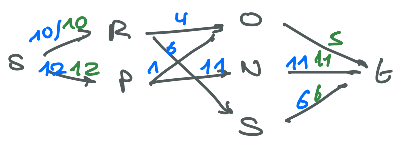
On enlève 800 à la valeur de toute solution.

Offre = demande ⇒ Dans toute solution, exacte^t 10 voitures partiront de R, 12 de P, et exacte^t 5 arrivent à Orléans, 11 à Rennes, et 6 à B

	O	N	B
R	0	200	50
P	0	175	100
		−175	−50

	O	N	B	
R	0	25	0	10
P	0	0	50	12
	5	11	6	

Sol° de coût 0 ? Graphe "des 0"



Flot de 22 unités

→ Sol° opt

	O	N	B
R	4		6
P	1	11	

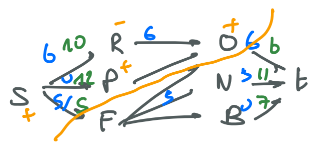
de coût $4\cdot 50 + 6\cdot 100 + 25 + 11\cdot 200 = 3025$

⇒ $50\times 10 + 25\cdot 12 + 175\cdot 11 + 50\cdot 6$.

	O	N	B		
R	50	250	100	−50	10
P	25	200	125	−25	12
F	0	0	0		5
	6	14	7		

Total : 27

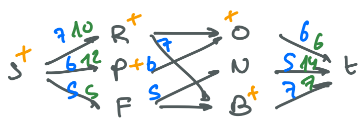
	O	N	B
R	0	200	50
P	0	175	100
F	0	0	0



arc qui rentre dans la coupe, qu'on utilise pas dans le flot max.

Sommets lignes marqués R et P : −50 Colonne marquée : O +50 δ = 50.

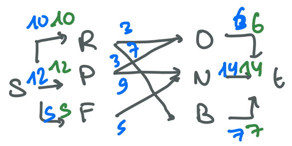
	O	N	B		
R	0	150	0	10	−125
P	0	125	50	12	−125
F	50	0	0	5	
	6	14	7		
	+125		+125		



lignes marqués R, P −125
colonnes marqué O, B +125

δ = 125

	O	N	B
R	0	25	0
P	0	0	50
F	175	0	125



La sol° optimale est

	O	N	B
R	3		7
P	3	9	
(F		5	)

de coût $3\cdot 50 + 7\cdot 100 + 3\cdot 25 + 9\cdot 200$
2725.

	O	N	B		
R	50	250	100	−50	10
P	25	200	125	−25	12
F	50	75	100		5
	6	14	7		

Exercice 59 (Problème de transport et PL)

On considère le problème de transport 3x3 dont la matrice des coûts est :

8	6	7
11	8	9
9	5	8

Les lignes correspondent à 3 dépôts D1, D2, D3 qui disposent de quantités 32, 15, 13 respectivement. Les colonnes correspondent à 3 clients dont les demandes sont de 26, 10 et 24 respectivement.

Q 59.1 Déterminez la solution dite du “coin Nord-Ouest” et indiquez précisément (en les présentant sous forme de tableau 3x3) les valeurs des différents flux dans cette solution, ainsi que le coût de cette solution.

Q 59.2 En appelant $x_{i,j}$ la quantité de produit transférée entre le dépôt i et le client j, exprimez le problème comme un programme linéaire en variables $x_{1,1}$, $x_{1,2}$, ... $x_{3,3}$ (9 variables en tout). Ce programme linéaire est-il sous forme standard ? Si la réponse est oui, expliquez pourquoi. Si elle est non, donnez la forme standard du problème.

Q 59.3 Ecrivez complètement le problème dual du programme linéaire sous forme standard obtenu à la question 2.

Q 59.4 Résolvez le problème de transport posé plus haut.

Q 59.5 En utilisant le théorème des écarts complémentaires, déduisez la solution optimale du problème dual obtenu à la question 3. Vérifiez l’égalité des valeurs optimales.

	C1	C2	C3	
D1	8	6	7-6	32
D2	11	8	9-8	15
D3	9	5	8-5	13
	26	10	24	

60 λ

60 μ

Offre = demande → équilibré.

1. Construction réalisable

	C1	C2	C3			
D1	26	6	0	~~32~~	~~6~~	0
D2	0	4	11	~~15~~	~~11~~	0
D3	0	0	13	~~13~~	0	
	~~26~~	~~10~~	~~24~~			
	0	~~4~~	~~13~~			
		0	0			

2. $\text{Min } z = 8x_{11} + 6x_{12} + 7x_{13} + 11x_{21} + 8x_{22} + 9x_{23} + 9x_{31} + 5x_{32} + 8x_{33}$

s.c.

$x_{11} + x_{12} + x_{13} = 32$ λ_1

$x_{21} + x_{22} + x_{23} = 15$ λ_2

$x_{31} + x_{32} + x_{33} = 13$ λ_3

$x_{11} + x_{21} + x_{31} = 26$ μ_1

$x_{12} + x_{22} + x_{32} = 10$ μ_2

$x_{12} + x_{23} + x_{33} = 24$ μ_3 car équilibre offre demande

$x_{ij} \geq 0,\ i = 1.2.3\ \ j = 1.2.3$

RQ. contrainte redondante

$\max\ 32\lambda_1 + 15\lambda_2 + 13\lambda_3 + 26\mu_1 + 10\mu_2 + 24\mu_3$

s.c.

$\lambda_1 + \mu_1 \leq 8$ (x_{11})

$\lambda_1 + \mu_2 \leq 6$ (x_{12})

$\lambda_1 + \mu_3 \leq 7$ (x_{13})

$\lambda_2 + \mu_1 \leq 11$

$\lambda_2 + \mu_2 \leq 8$

$\lambda_2 + \mu_3 \leq 9$

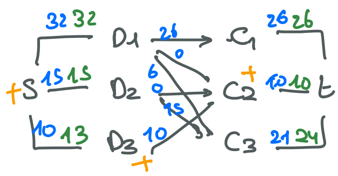
$\lambda_3 + \mu_1 \leq 9$

$\lambda_3 + \mu_2 \leq 5$

$\lambda_3 + \mu_3 \leq 8$

	C1	C2	C3
D1	2	0	1
D2	3	0	1
D3	4	0	3
	-2		-1

	C1	C2	C3	
D1	0	0	0	
D2	1	0	0	
D3	2	0	2	-2
		+2		

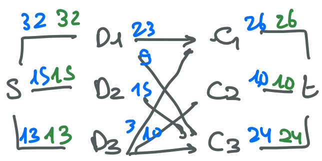


ligne marquée D3 -2

colonne marquée C2 +2

$\delta = 2$

	C1	C2	C3
D1	0	2	0
D2	1	2	0
D3	0	0	0



Sol° opt.

23	0	9
0	0	15
3	10	0

8	6	7	-6
11	8	9	-8
9	5	8	-5-2=-7
-2	+2-1		

$\lambda_1 = 6$ $\lambda_2 = 8$ $\lambda_3 = 7$

$\mu_1 = 2$ $\mu_3 = 2$ $\mu_2 = 1$

réalisable de (D)

optimales et valeur (TEC) Th écarts compl.

0	2	0
1	2	0
0	0	0

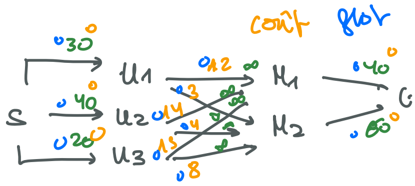
Exercice 56 (Résolution d'un problème de transport par Busacker et Gowen)

On considère un problème de transport entre 3 usines et 2 magasins. Les coûts c_{ij} de transport des produits de i vers j sont donnés ci-dessous :

	M1	M2	quantité disponible
U1	12	3	30
U2	14	4	40
U3	15	8	20
Besoin	30	60	

Q 56.1 Calculer la solution optimale de ce problème à l'aide de l'algorithme de Busacker et Gowen.

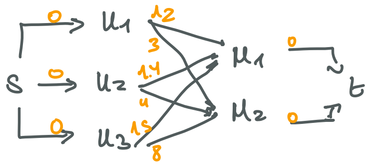
Réseau : arc : - capacité
- coût unitaire de transport



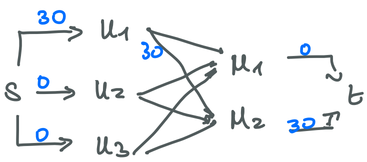
trouver un flot de 90 unités (flot max) à coût minimum

Algo de B&G ≈ F&F, mais chq étape : prendre un chemin de coût min dans le graphe d'écart

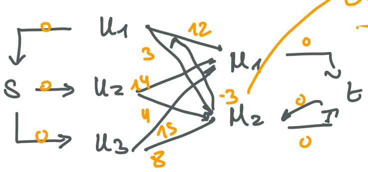
It 1 : graphe d'écart



Chemin de coût min
(s, U1, M2, t) coût 3
Capacité résiduelle 30
→ 30 sur (s, U1, M2, t)

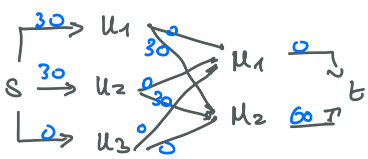


It 2 : graphe d'écart



diminution de flot
→ "rapporte 3"

Chemin de coût min
(s, U2, M2, t) : coût 4
Capacité résiduelle : 30



... jusqu'à un flot de 90 unités.