TD 8: Programmation Linéaire et Algorithmes de texte

1 Recherche en espace constant

1.1 Recherche d'un motif auto-maximal

Question 1 : **Correction** : Démonstration que $Period(u) = |u| - \pi(|u|)$. On note $n = |u|$.

$$\begin{aligned}
Period(u) &= min\{0 < k \leq n \mid \forall i \in [1, n-k]\ u[i] = u[i+k]\} \quad \text{par def. periode} \\
&= min\{0 < k \leq n \mid u[1, n-k] = u[k+1, n]\} \quad \text{par reformulation} \\
&= min\{0 < n-k' \leq n \mid u[1, k'] = u[n-k'+1, n]\} \quad \text{changement } n-k = k' \\
&= n - max\{0 \leq k' < n \mid u[1, k'] \sqsupset u[1, |u|]\} \quad \text{par reformulation } = \\
&= n - \pi(n) \quad \text{définition } \pi
\end{aligned}$$

Question 2 : **Correction** : Prenons déjà des exemples pour la notation $MaxSuf(w)$. Soit l'alphabet $\Sigma = \{a, b\}$ et fixons l'ordre $a < b$. $MaxSuf(bab) = bab$ et $MaxSuf(abba) = bba$. Si l'ordre est $a > b$ alors $MaxSuf(bab) = ab$ et $MaxSuf(abba) = abba$. Pour ces deux mots, pour l'un des ordres le mot est auto-maximal.

Prenons le mot $abaa$. Pour l'ordre $a > b$ alors $MaxSuf(abaa) = aa$, et pour l'ordre $a < b$, $MaxSuf(abaa) = baa$.

Question 3 : **Correction** : Soit w tels que $MaxSuf(w) = w$ et soit w_1 un préfixe de w, ainsi $w = w_1 w_2$.

Supposons par l'absurde que w_1 ne soit pas auto-maximal, c'est-à-dire qu'il existe $u_1 \sqsupseteq w_1$ tel quel $u_1 > w_1$. Comme u_1 est plus court que w_1, il existe nécessairement un indice i tel que $u_1[i] > w_1[i]$ et $u_1[1, i-1] = w_1[1, i-1]$. Mais en concaténant w_2 à u_1 on obtient $u_1 w_2 > w_1 w_2 = w$ et $u_1 w_2$ est bien un suffixe de w, ce qui contredit l'hypothèse que w est auto-maximal.

```
periode_naive(w,j) :=
pe := 1;
pour i de 2 à j faire
si w[i] ≠ w[i − pe] alors pe := i;
retourner pe;
```

Question 4 : **Correction** : À la fin de l'itération où $i = k$, nous montrons que $pe = Period(w[1, k])$.

Pour cela, il faut démontrer deux propriétés:

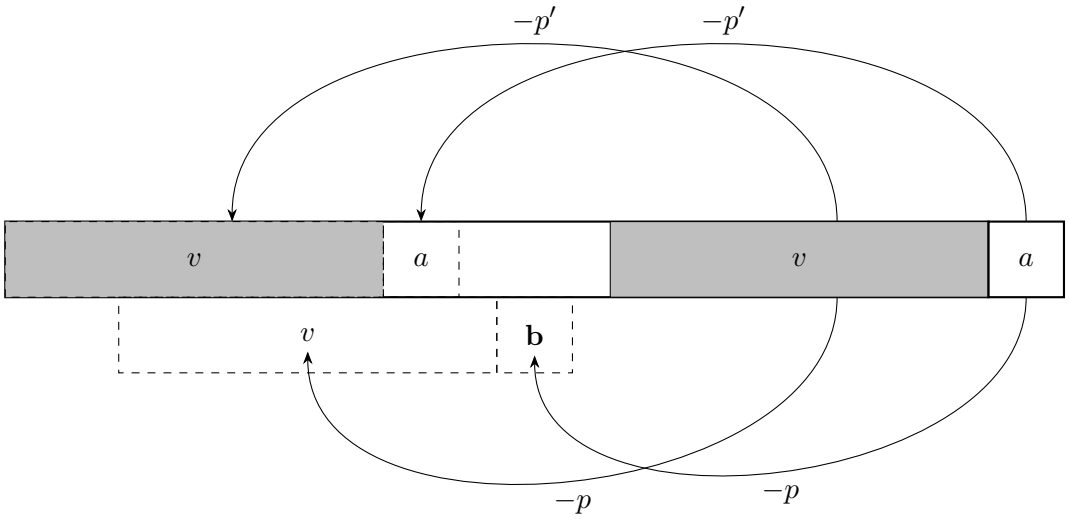
- Condition initiale: Avant de commencer la boucle **pour**, lorsque $i = 1$, on a $1 = Period(w[1, 1])$. En effet, un mot de longueur 1 a la période 1.
- Préservation: Si l'hypothèse est vrai en début de la i-ème itération de la boucle alors après la condition et l'exécution du corps de la $(i+1)$-éme boucle est vrai.

- Si $w[i] = w[i-pe]$ alors pe ne change pas et on démontre que pe est aussi période de $w[1,i]$ car pour tout $k \in [1, i-pe]$ $w[k] = w[k+pe]$ pour $k < i - pe$ (car $I(i)$) et $w[i-pe] = w[i]$. Par hypothèse d'induction, pe est minimal pour $w[1, i-1]$ donc aussi pour $w[1,i]$, donc pe est $Period(w[1,i])$, donc $I(i+1)$ est vrai.
- Si $w[i] \neq w[i-pe]$, notons $a = w[i]$, $b = w[i-pe]$.

  (A) On démontre que $a < b$ si w auto-maximal. Par hypothèse d'induction, $w[1, i-1] = w_1 b w_2$ avec $|w_1| = i - 1 - pe$ et $w_1 \sqsupset w[1, i-1]$. Donc $w[1,i] = w_1 b w_2 a$ et est auto-maximal car préfixe de w (propriété démontrée à la question précédente). Supposons par absurde que $a > b$. Comme w_1 est suffixe de $w[1, i-1]$, alors $w_1 a$ est un suffixe de $w[1,i]$, donc strictement supérieur à $w[1,i]$, contradiction avec $w[1,i]$ auto-maximal.

  (B) On démontre que $Period(w[1,i]) = i$.[1] Supposons par absurde que $p' = Period(w[1,i]) < i$. Alors p' est également une période de $w[1, i-1]$, donc $p' \geq p = Period(w[1, i-1])$. Mais comme $w[i-p] \neq w[i] = w[i-p']$ on déduit que $p' > p$. On suppose que $w[1, i-p'] = va$ et comme p' est une période de $w[1,i]$ alors $w[1,i] = w'_1 va$. Mais $p' > p = Period(w[1, i-1])$, alors $w[1, i-p'] \sqsubset w[1, i-1-p]$, donc $w[1, i-p] = w_1 vb = vaw_3$. Or $vaw_3 < vb$, contradiction avec $w[1, i-p]$ est auto-maximal (car préfixe de w). On déduit que $Period(w[1,i]) = i$, donc $I(i+1)$ est vrai après l'instruction conditionnelle.

Le schéma ci-dessous montre la seconde étape de la preuve, la duplication de va en vb par shift de $-p$ est toujours possible car $p < p'$ par hypothèse.



Question 5 : <u>Correction</u> : L'algorithme de Morris-Pratt utilise la fonction π pour calculer la position suivante dans le motif, j, quand $P[j+1] \neq T[i]$. Or la fonction π occupe $m = |P|$ entiers. Si on dispose de la période de $P[1,j]$ alors $\pi(j) = j - Period(P[1,j])$ (propriété démontré au TD précédent). Pour les motifs auto-maximaux, cette position est calculée avec periode_naive en $\Theta(j)$. On maintient donc *une* variable p contenant la période du préfixe courant $P[1,j]$, qui nécessite un recalcul lorsque p devient plus grand que j (sinon la période reste constante). Clairement, ce programme travaille avec un nombre fixé de variables.

```
MPAuto(P, T) : liste :=
```

[1] Une preuve plus courte utilise le résultat du TD 01 sur les périodes.

```
p := 1;
j := 0;
L := empty;
pour i de 1 à |T| faire
tant que j ≥ 0 ∧ P[j+1] ≠ T[i] faire
j := j − p;
si p > j alors p := periode_naive(P, j)
si P[j+1] = T[i] alors
j := j + 1;
si P[j] ≠ P[j − p] alors
p := j;
si j = |P| alors
AjoutListe(L, i−|P|)
j := j − p;
retourner L
```

On pose l'invariant en début de boucle: $p = Period(P[1,j])$ avec $Period(\epsilon) = 1$. Ainsi, dès que $j \geq 0$, nous avons $\pi(j) = j - p$ et si $j - p \geq p$ alors $Period(P[1, j-p]) = p$. De plus, si $P[j] \neq P[j-p]$, nous avons démontré (question précédente) que pour P auto-maximal p est j. Ces propriétés assurent que l'invariant est préservé dans la boucle principale, donc celle ci se comporte comme l'algorithme de Morris-Pratt.

Prouvons désormais la complexité en temps : dans la boucle **tant que**, à chaque décrémentation de la variable j d'un offset p, un recalcul a lieu, de complexité j, uniquement si $j < p$.

Rappel: Soit la fonction de potentiel *pot* définie sur les états de l'algorithme, la complexité amortie de l'opération *op* telle que $s \xrightarrow{op} s'$ est $a(op) = t(op) + pot(s') - pot(s)$. Pour un programme ayant une séquence d'opérations $\{op_i\}_{i\in[1,n]}$ en partant d'un état s_0, on obtient $\Sigma_{i=1}^{n} t(op_i) \leq \Sigma_{i=1}^{n} a(op_i) + pot(s_0)$.

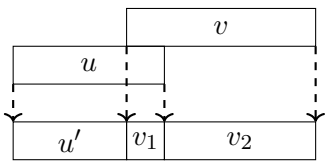
Soit $pot(s)$ est la valeur de j en s. Pour une itération de la boucle **tant que**, $a(op) = t(op) + (-p)$ où $t(op)$ est le coût de calcul de la période, donc $t(op) = j < p$, d'où $a(op) \leq 0$. Pour une itération de la boucle **pour**, $a(op') = t(op') + pot(s_{i+1}) - pot(s_i)$ où $t(op')$ est le coût des opérations dans le corps de la boucle, donc $t(op') \leq 1$ et j augmente d'au plus 1. Donc $a(op') \leq 2$. Alors le temps d'exécution de la boucle est $\leq 2n + pot(j_0) = 2n$. D'où la complexité $O(n)$ en temps, par analyse amortie.

Motivation for KMP: Nous déplaçons progressivement le motif le long du texte, en vérifiant dans quelle mesure il correspond au texte. L'idée principale est la suivante : supposons qu'à la position i du texte, il y ait une correspondance partielle entre le motif et le texte jusqu'à la position q du motif. Dans ce cas, nous déplacerons le début du texte vers cette position. Pourquoi peut-on sans risque ignorer les positions intermédiaires ?

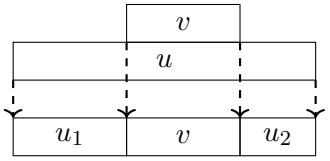
1.2 Recherche de texte en espace constant

Question 6 : <u>Correction</u> : Par l'absurde, si u apparaît avec un décalage $j \in \{i - |u| + 1, \ldots, i\}$, alors u commence avant v et fini après le début de v: u intersecte le début du mot v.

- Cas 1: u termine avant v. Alors $u = u'v_1$ et $v = v_1v_2$. Alors $v = v_1v_2 > v_2$ par auto-maximalité mais alors $v_1v_1v_2 = v_1v$ est un suffixe de w plus grand que $v_1v_2 = v = MaxSuf(w)$, absurde.



- Cas 2: u se termine après v. Alors $u = u_1 v u_2$, or $vu_2v > v$ (les deux mots commencent par v et le premier est plus long) ce qui est absurde.



Question 7 : **Correction** : TODO

2 LP avec matrices totalement unimodulaires

Question 8 : Soit $A \in \mathbb{Z}^{m\times n}$ totalement unimodulaire et non singulière, et $b \in \mathbb{Z}^m$. Une solution admissible $x \in \mathbb{R}^n$ du système $Ax \geq b$ est extrémale si et seulement si elle satisfait n inégalités linéairement indépendantes avec égalité. On peut donc extraire une sous-matrice carrée $A' \in \mathbb{Z}^{n\times n}$ et un vecteur $b' \in \mathbb{Z}^n$ tels que $A'x = b'$ et $\det(A') \neq 0$. Puisque A est totalement unimodulaire, on a $\det(A') = \pm 1$. Par la règle de Cramer, la solution s'écrit $x_i = \det(A'_i)/\det(A')$ pour $i = 1, \ldots, n$, où A'_i est obtenu en remplaçant la i-ème colonne de A' par b', ce qui montre que $x_i \in \mathbb{Z}$. Ainsi, toute solution extrémale du système $Ax \geq b$ est entière.

Note: The next 2 questions can be skipped since the matrix in the final question (for showing that there is an integer solution to the LP) has a simple form and we can make a simpler argument for that matrix.

Question 9 : **Correction** : Le fait que A possède une partition admissible a comme conséquence que pour toute paire de lignes (i_1, i_2) les *coefficients non nuls* de ces lignes sont, pour chaque colonne, soit tous du même signe (pour $i_1, i_2 \in I''$) soit tous de signes contraires (pour $i_1, i_2 \in I'$).

Comme la matrice A_1 est extraite de A et quelle a, sur toute colonne, exactement deux coefficients non nuls,

il résulte que le vecteur $\sum_{i\in I_1\cap I'} A_1[i,-]$ a des coefficients en $\{0,-1,1\}$ et les coefficients non nuls apparaissent sur des colonnes où le vecteur $\sum_{i\in I_1\cap I''} A_1[i,-]$ a des coefficients non nuls du même signe.

Supposons par absurde que le vecteur $\sum_{i\in I_1\cap I''} A_1[i,-]$ a des coefficients différents de $\{0,1,-1\}$, càd 2 ou -2. Alors il existe deux lignes $i_1, i_2 \in I_1 \cap I''$ telles que $\exists j \in J$ avec $A_1[i_1, j] = A_1[i_2, j]$, contradiction avec la condition i de la partition admissible.

En prenant $\lambda_i = 1$ pour $i \in I_1 \cap I'$ et $\lambda_i = -1$ pour $i \in I_2 \cap I''$, on obtient:

$$\sum_{i\in I_1} \lambda_i A_1[i,-] = \sum_{i\in I_1\cap I'} A_1[i,-] - \sum_{i\in I_1\cap I''} A_1[i,-] = 0 \tag{1}$$

Question 10 : **Correction** : Par récurrence sur la dimension d de A_1:
Cas $d = 1$: comme tout coefficient de A est en $\{0,1,-1\}$ et A_1 est extraite de A, alors $\det(A_1) = A_1[1,1] \in \{0,1,-1\}$.
Cas $d > 1$: On suppose que la propriété est vraie pour toute dimension $\leq d$.

a. Si A_1 a une colonne j avec tous les coefficients 0, alors $\det(A_1) = \sum_{i=1}^{d+1} A_1[i,j](-1)^{i+j} \det(A_{1,i,j}) = 0$.

b. S'il existe une colonne j de A_1 avec un seul coefficient non nul à la ligne i, alors $\det(A_1) = \underbrace{A_1[i,j]}_{\in\{0,1,-1\}} (-1)^{i+j} \underbrace{\det(A_{1,i,j})}_{\in\{0,1,-1\} \text{ par HR}}$ et donc $\det(A_1) \in \{0,1,-1\}$.

c. Si toutes les colonnes de A_1 ont deux coefficients non nuls. D'après l'exercice précédent, il existe une combinaison linéaire non nulle $\{\lambda_i\}_{i\in I_1}$ tel que $\sum_{i\in I_1}\lambda_i A_1[i,-]=0$. Soit $\ell\in I_1$ une ligne ayant au moins un coefficient non nul. On construit A_1' en multipliant la ligne ℓ de A_1 par λ_ℓ. Alors $\det(A_1')=\lambda_\ell\det(A_1)$.

On construit A_1'' en ajoutant à la ligne ℓ la combinaison linéaire des autres lignes en utilisant les coefficients $\{\lambda_i\}_{i\in I_1\setminus\{\ell\}}$. Alors A_1'' a que des coefficients nuls à la ligne ℓ et $\det(A_1'')=\det(A_1')$. De plus, une colonne de A_1'' au plus une valeur non nulle et en utilisant les points a et b on obtient que $\det(A_1'')\in\{0,1,-1\}$. Comme $\det(A_1')=\lambda_\ell\det(A_1)$ et $\lambda_\ell\in\{0,1,-1\}$, alors $\det(A_1)\in\{0,1,-1\}$.

Question 11 : **Correction** : Le problème admet la solution optimale suivante: $x_{i,j}=1$ pour tout $(i,j)\in E$ et $x_i=1$ si $i\in V\setminus\{t\}$ et $x_t=0$.

Question 12 : **Correction** : D'après l'équation (1), $x_{i,j}\geq x_i-x_j$ et comme les inconnues sont positives alors $x_{i,j}\geq 0$. Donc $x_{i,j}\geq\max(x_i-x_j,0)$ pour $\forall(i,j)\in E$.

D'après l'équation (2), $x_s>x_t$.

Supposons par absurde que, dans une solution optimale x, il existe $i\in V$ avec $x_\ell>x_s$. Soit la solution x' (voir Figure 2) telle que

- $x_i'=\max(x_i-(x_\ell-x_s),0)$ si i appartient à la composante fortement connexe S de ℓ où s'il existe un chemin de i vers S, et
- $x_i'=x_i$ sinon.

On peut démontrer facilement que $\sum_{(i,j)\in E}c(i,j)\cdot x_{i,j}'$ est diminuée, contradiction avec x solution optimale.

Supposons par absurde que, dans une solution optimale x, il existe un ℓ tel que $x_\ell<x_t$. Soit X l'ensemble de sommets ℓ tel que $x_\ell<x_t$. Soit la solution x' (voir Figure 2) telle que

- $x_i'=x_t$ si $i\in X$, et
- $x_i'=x_i$ sinon.

On peut encore démontrer facilement que $\sum_{(i,j)\in E}c(i,j)\cdot x_{i,j}'$ est diminuée, contradiction avec x solution optimale.

Question 13 : **Correction** : Supposons qu'il existe une solution optimale x avec $x_t>0$. On construit la solution x' en changeant $x_i'=x_i-x_t$ pour tout $i\in V$ et les valeurs de $x_{i,j}'=\max(x_i'-x_j',0)=x_{i,j}$. Donc la solution x' est aussi optimale et de plus $x_t'=0$.

Supposons qu'il existe une solution optimale x avec $x_s>1$. On construit une solution x' telle que $x_i'=\frac{x_i}{x_s}$ pour tout $i\in V$ et $x_{i,j}'=\max(x_i'-x_j',0)=\frac{\max(x_i-x_j,0)}{x_s}\leq x_{i,j}$. Donc x soit n'est pas optimale (contradiction), soit x était nulle (contradiction avec $x_s>1$).

Question 14 : **Correction** : En écrivant ce problème comme un problème PL canonique on obtient $c=(\cdots c_{i,j}\cdots 0\cdots 0)$, $x=(\cdots x_{i,j}\cdots x_1\cdots x_s\ x_t)$,

$$A=\left(\begin{array}{ccc|ccccc} 1 & \cdots & 0 & 1 & 0 & \cdots & -1 & 0\\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & 1 & 0 & -1 & \cdots & 0 & 1\\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & -1 \end{array}\right)$$

(donc à gauche une matrice identité $|E|^2$ sur une ligne 0, et à droite une matrice ayant les coefficients dans $\{0,1,-1\}$ qui est un exemple à part la dernière ligne), et $b=(0\cdots 0\ 1)$. On démontre que A est totalement unimodulaire.

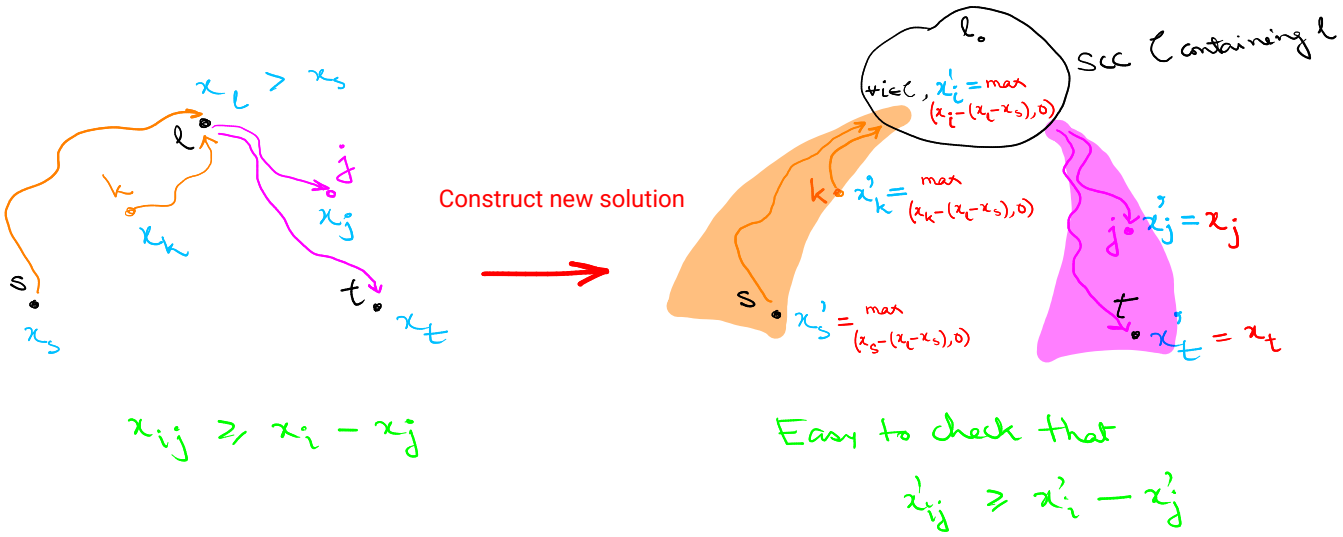


Figure 1: The case $x_\ell > x_s$ for some ℓ

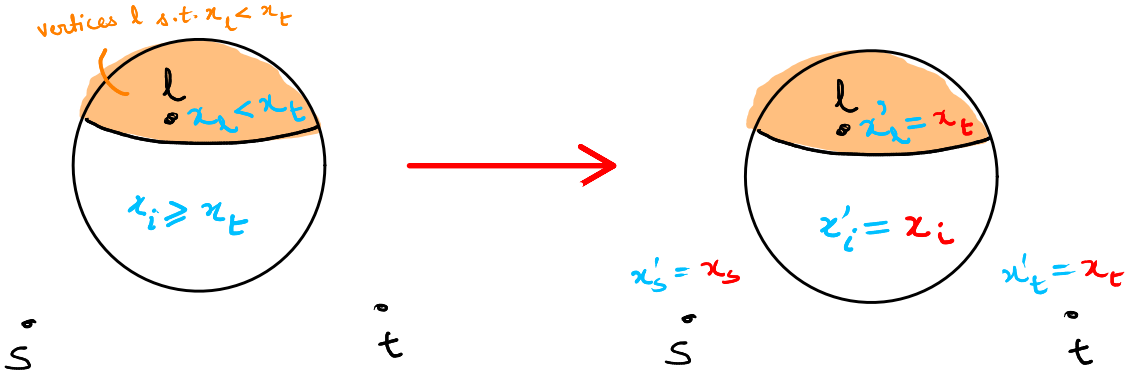


Figure 2: The case $x_\ell < x_t$ for some ℓ

- Si l'extraction de sous-matrice carrée est faite à l'intérieur de la partie gauche de A (matrice identité avec bordée par une ligne 0), le déterminant de cette extraction est soit 1 soit 0.
- Si l'extraction de sous-matrice est faite à l'intérieur de la matrice B, la sous-matrice de A de taille $(|E|+1) \times |V|$ bordant à droite A,

  on démontre que B est totalement unimodulaire. Comme G est connexe, alors $|V| \leq |E|+1$. Alors B^T est une matrice unimodulaire dont les $(|E|+1)$ colonnes ont exactement deux valeurs non nulles (colonne = arête dans E et contrainte $x_s - x_t \geq 1$). De plus, les indices de ligne de B^T admettent une partition admissible avec $I' = I$ et $I'' = \emptyset$ car deux sommets (lignes) ne peuvent pas apparaître avec le même signe (si non nuls) sur une arrête (colonne) dans les contraintes.

  On utilise la question 3 pour B^T et donc les sous-matrices carrées B_1 de B^T ont $\det(B_1) \in \{0, 1, -1\}$. Ainsi, la partie B de A est totalement unimodulaire.
- Si l'extraction de sous-matrice est faite en combinant des colonnes de la matrice identité bordée par une ligne 0 et de B, alors
  - si une colonne/ligne avec que des 0 est extraite de la matrice identité, le déterminant est 0,

– si toutes les colonnes extraites de la matrice identité ont un coefficient 1, alors on calcule le déterminant de cette extraction en utilisant ces colonnes jusqu'à les éliminer; la matrice qui reste est une extraction carrée de B, qui a comme déterminant une valeur dans $\{0, 1, -1\}$.

Donc A est totalement unimodulaire.

Alors, en appliquant la question 1 et le codage d'un problème standard en un problème canonique, on obtient que toutes les solutions sont entières. De plus, d'après la question précédente, les solutions pour x_i sont soit 0 soit 1.

Le problème LP code le problème du *min-cut*: tous les sommets i dont $x_i = 1$ appartiennent à l'ensemble $S \ni s$, les sommets j avec $x_j = 0$ appartiennent à l'ensemble $T \ni t$ avec $S \uplus T = V$. Les arêtes $(i, j) \in E$ telles que $x_{i,j} = 1$ correspondent à une arête avec le sommet i dans S ($x_i = 1$) et le sommet j dans T ($x_j = 0$), donc font partie de la coupure.

Question 15 : <u>Correction</u> : Rappel: dans le problème PL, I est l'ensemble des indices de lignes de A contraintes par des égalités, I' est celui des indices des lignes de A contraintes par des inégalités, et J est l'ensemble d'indices de colonnes de A (inconnues) qui sont positives.

La formulation de LP en problème PL correspond à $I = \emptyset$ et $|I'| = |E| + 1$, $|J| = |E| + |V|$. Donc le problème dual de LP se formule par:

$$\texttt{Maximiser } y \cdot b \texttt{ tel que}$$

$$\forall j \in J \; y \cdot A[-, j] \leq c_j \tag{2}$$

$$\forall i \in I' \; y_i \geq 0 \tag{3}$$

avec $y = (\underbrace{\cdots y_{i,j} \cdots}_{|E|} \; y_{t,s})$, $b = (\underbrace{\cdots 0 \cdots}_{|E|} \; 1)$ et $c_j = (\cdots c_{(i,j)\in E} \cdots \underbrace{0 \cdots 0}_{i \in V})$

Son interprétation comme un problème de graphe correspond à maximiser $y_{t,s}$ sous la contrainte que $y_{i,j} \leq c_{i,j}$ (car la partie gauche de A est la matrice identité) et $y_{i,j} \geq 0$. Donc c'est bien le flot maximal entre s et t.

Le théorème de dualité des problèmes PL dit que la solution optimale est la même, donc $\sum_{(i,j)\in E} c(i, j)x_{i,j} = y_{t,s}$, ce qui correspond au théorème de dualité entre flow-max et min-cut.