Corrigé : espace des cycles d'un graphe (ENS 2008)

Partie 1. Arbres couvrants et cycles fondamentaux

Question 1.1

1. (a) Dans un cycle élémentaire les sommets $s_1, s_2, \ldots, s_\lambda$ sont deux à deux distincts donc $\lambda \leqslant n$. On peut donc majorer (très grossièrement) le nombre de cycles élémentaires par $\displaystyle\sum_{\lambda=3}^{n} \frac{n!}{(n-\lambda)!}$ qui est une quantité finie.

(b) Considérons un graphe $G' = (S, A')$ connexe tel que $A' \subset A$ et A' est de cardinal minimal. S'il contenait un cycle, on pourrait supprimer une arête de ce dernier sans perdre le caractère connexe, et ainsi contredire l'hypothèse de minimalité de $|A'|$. G′ ne contient donc pas de cycle ; il s'agit bien d'un arbre couvrant.

2. Montrons par récurrence sur n qu'un arbre $T = (S, A)$ (*ie* un graphe acyclique et connexe) à n sommets possède $n-1$ arêtes.

- Le résultat est évident lorsque $n = 1$.
- Si $n > 0$, supposons le résultat acquis au rang $n-1$. T est connexe donc tout sommet est au moins de degré 1.
  - Supposons l'existence d'un sommet s de degré 1. il existe un unique sommet t tel que $(s,t) \in A$. Posons alors $T' = (S \setminus \{s\}, A \setminus \{(s,t)\})$. T′ est toujours acyclique et connexe donc par hypothèse de récurrence possède $n-2$ arêtes. L'arbre T possède donc $n-1$ arêtes.
  - Supposons que tous les sommets soient au moins de degré 2. Partant d'un sommet $s_0 \in S$, on peut construire une suite $(s_i)_{i\in\mathbb{N}}$ tel que pour tout $i \geqslant 1$, $(s_{i-1}, s_i) \in A$ et $(s_i, s_{i+1}) \in A$. Mais d'après le principe des tiroirs, dans $s_0, s_1, \ldots, s_n$ se trouve forcément un cycle, ce qui est absurde.

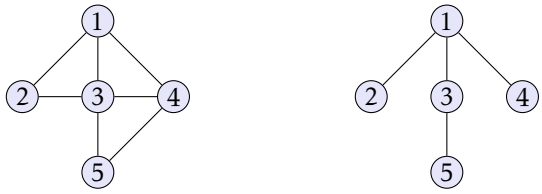
3. Pour déterminer la présence d'un cycle, on effectue un parcours (en profondeur ou en largeur) à partir d'un sommet s_0 quelconque (par exemple $s_0 = 1$) en mémorisant, lors de la découverte d'un sommet, le sommet qui l'a « découvert » (autrement dit son père dans l'arbre enraciné en s_0). Puisque G est connexe, tous les sommets sont accessibles et si, lors du parcours, on rencontre un sommet déjà vu et autre que son père, c'est que le graphe présente un cycle.

```
fonction SansCycleElem(Adj)
    pour i de 1 à n faire
        père[i] ← 0
    père[1] ← 1
    àTraiter ← 1
    tant que àTraiter ≠ ∅ faire
        àTraiter → s
        pour t ∈ Adj[s] faire
            si père[t] > 0 et père[t] ≠ s alors
                renvoyer Faux
            sinon
                père[t] ← s
                àTraiter ← t
    renvoyer Vrai
```

Si les accès au tableau père et à la pile (ou file) àTraiter sont de coût constant, cette fonction possède une complexité en $O(n+m)$: la création du tableau des pères se réalise en $O(n)$, et dans le pire des cas chaque sommet entre puis sort de àTraiter (pour une complexité en $O(n)$) et chaque arête est parcourue une fois (pour une complexité en $O(m)$).

Question 1.2

1. Un dessin vaut mieux qu'un long discours.



$T = (S, A')$ avec $A' = \{(1,2),(1,3),(1,4),(3,5)\}$ est un arbre couvrant ; sa structure d'adjacence est le tableau

`[[2, 3, 4], [1], [1, 5], [1], [3]]`.

2. Un parcours (en largeur ou en profondeur) rencontre les sommets d'un graphe en empruntant les arêtes d'un arbre couvrant. D'où la fonction :

```
fonction ArbreCouvrant(Adj)
    pour i de 1 à n faire
        déjàVu[i] ← Faux
        Adj'[i] ← []
    déjàVu[1] ← Vrai
    àTraiter ← 1
    tant que àTraiter ≠ ∅ faire
        àTraiter → s
        pour t ∈ Adj[s] faire
            si déjàVu[t] = Faux alors
                Adj'[s] ← t :: Adj'[s]
                Adj'[t] ← s :: Adj'[t]
                déjàVu[t] ← Vrai
                àTraiter ← t
    renvoyer Adj'
```

Pour les mêmes raisons qu'à la question 1.1.3, la complexité de cette fonction est en $O(n+m)$.

Question 1.3

1. Ajouter l'arête a ne fait pas perdre à T son caractère connexe. En revanche, puisqu'il y a maintenant n arêtes et non plus $n-1$, la question 1.1.2 montre qu'il ne s'agit plus d'un arbre. C'est donc qu'il y a eu apparition d'un cycle élémentaire, et ceci prouve l'existence de φ_a.

Supposons l'existence d'un autre cycle élémentaire φ'_a créé par l'ajout de cette arête. Puisque $a = (s,t)$, c'est qu'il existe dans T deux chaînes élémentaires distinctes reliant s à t. Mais ces deux chaînes forment dans T un cycle, ce qui est exclus par hypothèse. D'où l'unicité de φ_a.

2. (a) On peut majorer le nombre d'arêtes $a = (s,t) \in A \setminus A'$ par le nombre total d'arêtes possibles, soit $\binom{n}{2} = \dfrac{n(n-1)}{2}$.

Chacun des cycles élémentaires φ_a potentiellement créé est de longueur inférieure ou égale à n, ce qui conduit à la majoration : $L_T \leqslant \dfrac{n^2(n-1)}{2}$

(b) Il s'agit ici de calculer L_T pour différents arbres couvrants du graphe complet (puisque $m = \dfrac{n(n-1)}{2}$).

Considérons tout d'abord l'arbre $T_1 = (S, A_1)$ avec $A_1 = \left\{(1,i) \;\middle|\; 2 \leqslant i \leqslant n\right\}$. Les arêtes de $A \setminus A_1$ sont de la forme $a = (i,j)$ avec $2 \leqslant i < j \leqslant n$ et le cycle φ_a est de longueur 3. On a donc $L_{T_1} = 3\binom{n-1}{2} = \dfrac{3(n-1)(n-2)}{2} = \Theta(n^2)$.

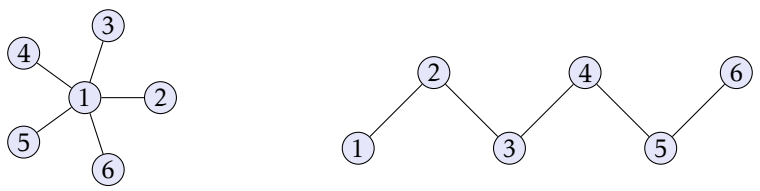


Figure 1 – Les arbres T_1 et T_2 pour $n = 6$.

Considérons maintenant l'arbre $T_2 = (S, A_2)$ avec $A_2 = \left\{(i,i+1) \;\middle|\; 1 \leqslant i \leqslant n-1\right\}$. Les arêtes de $A \setminus A_2$ sont de la forme $a = (i, i+k)$ avec $k \geqslant 2$ et $1 \leqslant i \leqslant n-k$, et le cycle φ_a est de longueur $k+1$. On a donc :

$$L_{T_2} = \sum_{k=2}^{n-1} (k+1)(n-k) = (\text{calcul}) = \frac{(n-2)(n-1)(n+6)}{6} = \Theta(n^3).$$

L'ordre de grandeur de L_T dépend donc de T.

3. (a) Revenons sur le pseudo-programme **`ArbreCouvrant`** défini à la question 1.2.2. Lors de l'exploration des voisins du sommet s, chaque sommet t qui a déjà été vu correspond à une arête $a = (s,t)$ fermant un cycle dans l'arbre T. Dès lors, il suffit de calculer le chemin qui les relie dans T pour connaître φ_a. Notons qu'il n'est pas nécessaire d'attendre que la construction de T soit achevée ; à cet instant ce chemin est déjà présent dans T.

Nous allons donc commencer par rédiger une fonction **`Cycle`** qui calcule le chemin qui relie deux sommets s et t dans un arbre T. Nous allons réaliser un parcours (en largeur ou en profondeur, peu importe) à partir de s (vu comme racine de l'arbre orienté associé), en notant dans un tableau le père de chaque sommet. Une fois t découvert, ce tableau permettra de reconstituer le chemin de s à t et donc le cycle φ_a.

```
fonction Cycle(Adj, s, t)
    pour i de 1 à n faire
        père[i] ← 0
    père[s] ← s
    àTraiter ← s
    tant que père[t] = 0 faire
        àTraiter → u
        pour v ∈ Adj[u] faire
            père[v] ← u
            àTraiter ← v
    x ← t
    c ← [t]
    tant que x ≠ s faire
        x ← père[x]
        c ← x :: c
    renvoyer t :: c
```

```
fonction Phi(Adj)
    pour i de 1 à n faire
        déjàVu[i] ← Faux
        Adj'[i] ← []
    déjàVu[1] ← Vrai
    àTraiter ← 1
    Phi ← ∅
    tant que àTraiter ≠ ∅ faire
        àTraiter → s
        pour t ∈ Adj[s] faire
            si déjàVu[t] = Faux alors
                Adj'[s] ← t :: Adj'[s]
                Adj'[t] ← s :: Adj'[t]
                déjàVu[t] ← Vrai
                àTraiter ← t
            sinon
                Phi ← Phi ∪ Cycle(Adj', s, t)
    renvoyer Phi
```

(b) Dans le cas d'un arbre, un parcours a pour complexité $O(n)$ puisqu'il possède $n-1$ arêtes. La complexité de la fonction **`Cycle`** est donc en $O(n)$.

Dans le cas général, le parcours d'un graphe a une complexité en $O(n+m)$, donc on peut majorer la complexité de la fonction **`Phi`** par un $O(n+m+n\operatorname{card}\varphi(G,T))$. Sans doute peut-on abaisser ce coût en un $O(n+m+L_T)$ au prix d'une complexité spatiale plus importante.

Partie 2. Bases minimales de l'espace des cycles

Question 2.1

1. (a) Si $\alpha = 0$, $\alpha e = \{\}$; si $\alpha = 1$, $\alpha e = e$.

Dans $\mathbb{K}$ on a $-1 = 1$ donc $-e = e$.

Enfin, sachant que dans $\mathbb{K}$ on a $0+0 = 0$, $1+0 = 1$ et $1+1 = 0$, $e+f$ représente la différence symétrique des sous-ensembles e et f : $e+f = (e \cup f) \setminus (e \cap f)$.

(b) $\langle e \mid f \rangle = 1 \iff \operatorname{card}\left\{i \in [\![1,m]\!] \;\middle|\; e_i f_i = 1\right\}$ est impair $\iff \operatorname{card}\left\{i \in [\![1,m]\!] \;\middle|\; e_i = f_i = 1\right\}$ est impair

$\iff \operatorname{card}\left\{i \in [\![1,m]\!] \;\middle|\; a_i \in e \text{ et } a_i \in f\right\}$ est impair $\iff \operatorname{card} e \cap f$ est impair.

2. S est non vide donc contient au moins une arête $a \in A \setminus A'$. Posons $C = \varphi_a$. À l'exception de l'arête a, toutes les autres arêtes qui composent le cycle φ_a appartiennent à A', donc $S \cap \varphi_a = \{a\}$. D'après la question précédente, $\langle C \mid S \rangle = 1$.

3. Nous avons $\varphi(G,T) = \left\{\varphi_{a_1}, \dots, \varphi_{a_N}\right\}$.

- Montrons que $\varphi(G,T)$ est une famille libre.

  Soit $(\lambda_1, \dots, \lambda_N) \in \mathbb{K}^N$ tel que $\displaystyle\sum_{i=1}^{N} \lambda_i \varphi_{a_i} = 0$. Pour tout $j \in [\![1,N]\!]$ on a $\operatorname{card}\{a_j\} \cap \varphi_{a_j} = 1$ et $\operatorname{card}\{a_j\} \cap \varphi_{a_i} = 0$ si $i \neq j$

  donc $\langle \{a_j\} \mid \varphi_{a_i} \rangle = \delta_{ij}$ et ainsi $\lambda_j = \left\langle \{a_j\} \;\middle|\; \displaystyle\sum_{i=1}^{N} \lambda_i \varphi_{a_i} \right\rangle = \langle \{a_j\} \mid 0 \rangle = 0$.

- Montrons que $\varphi(G,T)$ est une famille génératrice. Il suffit pour cela de montrer qu'elle engendre tout cycle C de G. À cet effet, posons $C' = C + \displaystyle\sum_{a \in C \setminus A'} \varphi_a$.

Pour tout $a \in A \setminus A'$, toutes les arêtes de φ_a à l'exception de a sont dans A'. De ceci il résulte que $a \notin C'$. En effet, c'est évident si $a \notin C$, et si $a \in C$ alors, puisque l'addition agit comme la différence symétrique, on a encore $a \notin C'$. Ainsi, $C' \subset A'$. Mais A' ne contient aucun cycle, donc $C' = \{\}$, et $C = \sum_{a \in C \setminus A'} \varphi_a$.

Question 2.2

1. Supposons l'existence de scalaires $(\lambda_1, \ldots, \lambda_{i-1}) \in \mathbb{K}^{i-1}$ tels que $C_i = \sum_{k=1}^{i-1} \lambda_k C_k$. En effectuant le produit scalaire de cette égalité par S_i on obtient l'égalité $1 = 0$ dans $\mathbb{K}$, ce qui est absurde. C_i est donc linéairement indépendant de $\{C_1, \ldots, C_{i-1}\}$.

2. De la question précédente il résulte que $\{C_1, \ldots, C_N\}$ est une famille libre de $\mathcal{C}(G)$, donc une base puisque de cardinal N. Nous allons maintenant prouver l'invariant suivant : pour tout $i \in [\![0, N]\!]$, $\{C_1, \ldots, C_i\}$ est inclus dans une base minimale de $\mathcal{C}(G)$.

- C'est clair pour $i = 0$.
- Si $i > 0$, supposons que $\{C_1, \ldots, C_{i-1}\}$ soit inclus dans une base minimale B, et décomposons C_i dans cette base : $C_i = B_1 + B_2 + \cdots + B_k$. Puisque $\langle C_i \mid S_i \rangle = 1$, il existe j tel que $\langle B_j \mid S_i \rangle = 1$, et par définition de C_i on a $w(B_j) \geqslant w(C_i)$. Par ailleurs, par définition de S_i, B_j ne peut être égal à aucun des C_k pour $1 \leqslant k \leqslant i-1$ puisque $\langle S_i \mid C_k \rangle = 0$.
  Posons alors $B' = B \setminus \{B_j\} \cup \{C_i\}$. B' est une base minimale puisque $w(B') \leqslant w(B)$ et elle contient $\{C_1, \ldots, C_i\}$, ce qui prouve la validité de l'invariant.

Question 2.3

1. Raisonnons matriciellement dans la base des $\{a_j\}$: les vecteurs $(C_1, \ldots C_{i-1})$ sont représentés par une matrice $M \in \mathbb{K}^{m \times (i-1)} = \begin{pmatrix} M_1 \\ M_2 \end{pmatrix}$ avec $M_1 \in \mathbb{K}^{N \times (i-1)}$ et S_i par un vecteur $Y = \begin{pmatrix} X \\ O \end{pmatrix} \in \mathbb{K}^m$ avec $X \in \mathbb{K}^N$, et on cherche à trouver $X \neq 0$ tel que $M^T Y = 0$, soit $M_1^T X = 0$. Ce système linéaire est constitué de $(i-1)$ équations et de N inconnues, ce qui assure l'existence d'une solution non nulle.

On obtient cette solution en employant la méthode du pivot de Gauss, de complexité $O(N^3)$. Le coût total pour le calcul de l'ensemble des supports est donc un $O(mN^3) = O(m^4)$.

2. (a) Observons que si on dispose de cette famille $(R_j^{(i)})$, il suffit de poser $S_i = R_i^{(i)}$ pour obtenir un support vérifiant les conditions requises par l'algorithme **`BaseMinimale1`**.

Pour construire cette famille, on commence par poser $R_1^{(1)} = \{a_1\}, \ldots, R_N^{(1)} = \{a_N\}$. Ces vecteurs sont linéairement indépendants.

Supposons maintenant construits $R_i^{(i)}, \ldots, R_N^{(i)}$. On pose $S_i = R_i^{(i)}$ puis on construit C_i. On définit ensuite les vecteurs $R_{i+1}^{(i+1)}, \ldots, R_N^{(i+1)}$ comme suit :

$$R_j^{(i+1)} = \begin{cases} R_j^{(i)} & \text{si } \langle C_i \mid R_j^{(i)} \rangle = 0 \\ R_i^{(i)} + R_j^{(i)} & \text{si } \langle C_i \mid R_j^{(i)} \rangle = 1 \end{cases}$$

Ces nouveaux vecteurs $R_{i+1}^{(i+1)}, \ldots, R_N^{(i+1)}$ sont bien linéairement indépendants, on a toujours $\langle C_k \mid R_j^{(i+1)} \rangle = 0$ pour $k \leqslant i-1$ puisque $\langle C_k \mid R_j^{(i)} \rangle = \langle C_k \mid R_i^{(i)} \rangle = 0$, et enfin $\langle C_i \mid R_j^{(i+1)} \rangle = 0$ puisque $\langle C_i \mid R_i^{(i)} \rangle = \langle C_i \mid S_i \rangle = 1$.

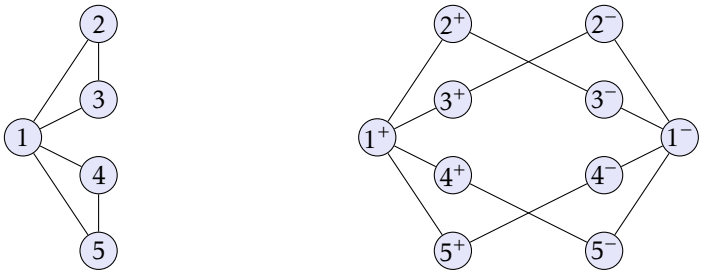
(b) On peut donc réécrire l'algorithme **`BaseMinimale1`** comme suit :

```
pour i de 1 à N faire
    R_i ← {a_i}
pour i de 1 à N faire
    S_i ← R_i
    C_i ← un cycle de G de poids minimal tel que ⟨C_i | S_i⟩ = 1
    pour j de i + 1 à N faire
        si ⟨C_i | R_j⟩ = 1 alors
            R_j ← S_i + R_j
```

Les opérations vectorielles (somme et produit scalaire) sont de complexité linéaire $O(m)$ donc, en faisant abstraction du coût du calcul des C_i, cette fonction est de complexité $O(N^2 m) = O(m^3)$.

Question 2.4

1. Les graphes G et G_i sont les suivants :



2. (a) Au chemin σ de G_i on fait correspondre le chemin σ' de G en identifiant chaque sommet t^+ ou t^- de cette chaîne au sommet t de G. Par exemple, si $\sigma = (2^+, 3^-, 1^-, 4^-, 5^+, 1^+, 3^+, 2^-)$ dans l'exemple précédent, alors $\sigma' = (2, 3, 1, 4, 5, 1, 3, 2)$. On obtient pas encore un cycle de G car certaines arêtes peuvent apparaître plusieurs fois (lorsque les arêtes (s^+, t^-) et (s^-, t^+) sont présentes dans σ, par exemple). Dans ce cas, on supprime la première et la dernière occurence de ces arêtes, jusqu'à ne plus en garder que zéro ou une occurrence. Dans le cas de notre exemple, on supprime les arêtes $(2, 3)$ et $(3, 2)$, puis les arêtes $(3, 1)$ et $(1, 3)$, jusqu'à obtenir le cycle $C = (1, 4, 5, 1)$, de poids inférieur ou égal au poids de σ (puisque toutes les arêtes sont de poids positif ou nul).
Il reste à justifier que $C \neq \{\}$, ce qu'on va montrer en calculant $\langle C \mid S_i \rangle$. Si on note (G^+, G^-) la partition de G_i formée des sommets t^+ d'une part et des sommets t^- d'autre part, le chemin σ va de G^+ à G^- donc utilise un nombre *impair* d'arêtes reliant l'une de ces deux composantes à l'autre. Or par construction ces arêtes correspondent à des arêtes de S_i dans la réduction de σ à σ'. Il y a donc dans σ' un nombre *impair* d'arêtes appartenant à S_i, et sa réduction à C va donc encore en garder un nombre impair. Ainsi, on aura $\langle C \mid S_i \rangle = 1$.

(b) Pour obtenir le vecteur d'incidence, il suffit d'additionner dans l'espace vectoriel E les vecteurs $\{a_k\}$ présents dans la chaîne, puisque les arêtes supprimées le sont par paire.

```
fonction INCIDENCE(chaine)
    λ ← longueur(chaine)
    pour k de 1 à m faire
        C[k] ← 0
    pour j de 1 à λ faire
        k ← chaine[j]
        C[k] ← (C[k] + 1) mod 2
    renvoyer C
```

Cette fonction est de complexité $O(\lambda + m)$.

(c) Soit $D \in \mathcal{C}(G)$ tel que $\langle D \mid S_i \rangle = 1$ de poids minimal. D est un cycle élémentaire car si D était la réunion de plusieurs cycles élémentaires disjoints, l'un de ces cycles C_j au moins vérifierait $\langle C_j \mid S_i \rangle = 1$ (rappelons que cette égalité traduit le fait que D_j possède un nombre *impair* d'arêtes de S_i) et contredirait le fait que D est de poids minimal. Posons alors $D = (s_0, s_1, \ldots, s_\lambda)$ et faisons-lui correspondre le chemin $\sigma = (s'_0, \ldots, s'_\lambda)$ de G_i en posant $s'_0 = s_0^+$ et

$$s'_j = \begin{cases} s_j^+ & \text{si } s'_{j-1} = s_{j-1}^- \text{ et } (s_{j-1}, s_j) \in S_i \text{ ou si } s'_{j-1} = s_{j-1}^+ \text{ et } (s_{j-1}, s_j) \notin S_i \\ s_j^- & \text{si } s'_{j-1} = s_{j-1}^+ \text{ et } (s_{j-1}, s_j) \in S_i \text{ ou si } s'_{j-1} = s_{j-1}^- \text{ et } (s_{j-1}, s_j) \notin S_i \end{cases}$$

On obtient ainsi un chemin σ dans G_i de même poids que D. De plus, puisque $\langle D \mid S_i \rangle = 1$ le nombre de passages entre G^- et G^+ est *impair*, et donc $s'_\lambda = s_\lambda^- = s_0^-$. Et si σ n'était pas de poids minimal on pourrait en construisant C contredire le caractère minimal de D.

3. (a)

```
fonction CYCLEMINIMAL(G = (S, E), S_i)
    distMin ← +∞
    pour s ∈ S faire
        (distance, chaine) ← PlusCourteChaine(s)
        si distance < distMin alors
            distMin ← distance
            C ← INCIDENCE(chaine)
    renvoyer C
```

La fonction `PlusCourteChaine` est appelée n fois et la fonction INCIDENCE au plus n fois donc le coût total de cette fonction est en $O(n(m + n \log n))$.

(b) Puisque nous avons à calculer N cycles, le coût total des calculs des cycles de poids minimaux est en $O(Nn(m + n \log n))$ avec $N = m - n + 1$, soit un $O(mn(m + n \log n))$.
Compte tenu de la question 2.3.2b, le coût de l'algorithme `BaseMinimale1` est en $O(m^3 + mn(m + n \log n)) = O(m^3 + mn^2 \log n)$.