# Des marches dans des graphes

Épreuve pratique d'algorithmique et de programmation
Concours commun des Écoles normales supérieures

Durée de l'épreuve : 3 heures 30 minutes

Juin/Juillet 2023

**ATTENTION !**

N'oubliez en aucun cas de recopier votre $u_0$
à l'emplacement prévu sur votre fiche réponse

**Important.**

Il vous a été donné un numéro $u_0$ qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion !). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de $u_0$. Vous indiquerez vos réponses (correspondant à votre $u_0$) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures !
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre $n$, on demande l'ordre de grandeur en fonction du paramètre, par exemple : $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***.
Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

# 1 Préliminaires

## 1.1 Notations

On rappelle que pour deux entiers naturels $a$ et $b$, $(a \mod b)$ désigne le reste de la division entière de $a$ par $b$, c'est-à-dire l'unique entier $r$ avec $0 \leq r < b$ tel que $a = k \times b + r$ pour $k \in \mathbb{N}$. Ce sujet portant sur des graphes pondérés orientés, chaque utilisation du mot graphe doit être entendue comme graphe pondéré orienté. Un graphe est donc, pour ce sujet, un triplet $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$ où $\mathcal{N}$ est l'ensemble fini des sommets ou nœuds du graphe ; $\mathcal{E} \subset \mathcal{N}^2$ est l'ensemble des arcs, chaque arc étant une paire de sommets ; et $\mathsf{poids} : \mathcal{N} \to \mathbb{N}$ est une fonction associant à chaque sommet du graphe un poids positif. Notez que cette définition interdit les arcs multiples (deux sommets ne peuvent être reliés que zéro ou une fois), mais qu'il est possible d'avoir un arc d'un sommet $s$ à lui-même, par l'arc $(s, s)$. Un arc $(s, t)$ de $\mathcal{E}$ est appelé un arc sortant de $s$, et le sommet $t$ est un successeur de $s$. On notera $\mathsf{succ}_\mathcal{G}(s)$ l'ensemble des successeurs de $s$ dans $\mathcal{G}$ :

$$\mathsf{succ}_\mathcal{G}(s) = \{t \mid (s, t) \in \mathcal{E}\}$$

Par souci de simplicité, dans ce sujet, $\mathcal{N}$ est toujours constitué des nombres de 0 à $n - 1$ avec $n$ le nombre de sommets.

## 1.2 Génération de nombres pseudo-aléatoires

Étant donné $u_0$, on définit la récurrence :

$$\forall t \in \mathbb{N}, \quad u_{t+1} = (909\,091 \times u_t) \mod 1\,010\,101\,039$$

L'entier $u_0$ vous est donné, et doit être recopié sur votre fiche réponse avec vos résultats. Une fiche réponse type vous est donnée en exemple, et contient tous les résultats attendus pour une valeur de $u_0$ différente de la vôtre (notée $\widetilde{u_0}$). Il vous est conseillé de tester vos algorithmes avec cet $\widetilde{u_0}$ et de comparer avec la fiche de résultats fournie. Pour chaque calcul demandé, avec le bon choix d'algorithme le calcul ne devrait demander qu'au plus de l'ordre de quelques secondes, jamais plus d'une minute.
On aura souvent besoin de nombreuses valeurs consécutives de la suite $u_n$. Il est donc conseillé que votre implémentation calcule le tableaux de tous les $u_n$ jusqu'à un certain rang.

**Question 1** *Calculer les valeurs suivantes :*

***a)*** $u_1$ *mod* 1000 ***b)*** $u_{12}$ *mod* 1000

***c)*** $u_{1\,234}$ *mod* 1000 ***d)*** $u_{7\,654\,321}$ *mod* 1000

## 1.3 Extraction d'une sous-liste croissante

Soit $L$ une liste $(a_1, \ldots, a_n)$ d'entiers positifs. On note $\mathsf{incr\text{-}list}(L)$ la sous-liste de $L$ obtenue en ne gardant un $a_i$ que si il est plus grand que tous les éléments précédents dans la liste. Par exemple, $\mathsf{incr\text{-}list}(1, 4, 2, 3, 6, 5) = (1, 4, 6)$, puisque 2 et 3 ont été exclus car plus petits que 4 (qui les précède), et de même 5 a été exclu car plus petit que 6.
Formellement, si $L = (a_1, \ldots, a_n)$ alors $\mathsf{incr\text{-}list}(L) = (b_1, \ldots, b_k)$ telle que $(b_1, \ldots, b_k)$ est une liste d'entiers strictement croissante ($b_i < b_j$ quand $i < j$) et :

$$\{b_1, \ldots, b_k\} = \{a_i \mid \forall j < i,\, a_j < a_i\}$$

**Question 2** *Soit $v_n$ la suite telle que $\forall n \in \mathbb{N}$, $v_n = u_n \mod 9$. On pose $L_n$ la liste $(v_0, \ldots, v_{n-1})$. Implémenter la fonction incr-list$(\cdot)$, et utiliser votre implémentation pour calculer la somme des entiers de incr-list$(L_n)$, c'est à dire $\left(\sum_{s \in \textit{incr-list}(L_n)} s\right)$ pour les valeurs de $n$ suivants :*

***a)*** $n = 3$ ***b)*** $n = 6$ ***c)*** $n = 123$ ***d)*** $n = 1\,234$

Pour tester votre code, vous pouvez vous aider du fait que pour $\widetilde{u_0}$ on a $L_6 = (5, 5, 4, 3, 7, 2)$ et donc incr-list$(L_6) = (5, 7)$.

**Question à développer pendant l'oral 1** *Donner la complexité en temps de votre algorithme.*

### 1.4 Génération de graphe

Étant donné trois entiers strictement positifs $n$, $M$ et $p$, on note $G(n, M, p)$ le graphe $(\mathcal{N}, \mathcal{E}, \mathsf{poids})$ où $\mathcal{N} = \{0, \ldots, n-1\}$ et pour chaque sommet $s \in \mathcal{N}$ :

— La liste des arcs sortants potentiels du sommet $s$ est la liste :

$$(s, t_0), \ldots, (s, t_{M-1}) \qquad \text{où } \forall 0 \leq i < M,\ t_i = u_{s \times M + i} \mod n.$$

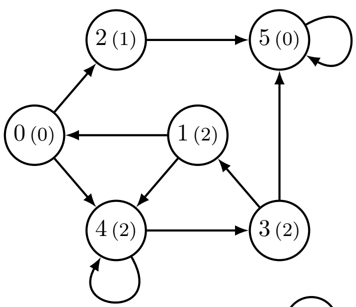
La liste des arcs sortants de $s$ est obtenue en ne gardant dans $(s, t_0), \ldots, (s, t_{M-1})$ que les arcs vers des sommets croissants. Plus précisément,

$$\mathsf{succ}_{\mathcal{G}}(s) = \left\{ t_j \mid t_j \in \mathsf{incr\text{-}list}(t_0, \ldots, t_{M-1}) \right\}.$$

On remarquera que tous les sommets ont au plus $M$ successeurs. De plus, puisque $(s, t_0)$ est toujours un arc sortant de $s$, tous les sommets ont au moins 1 successeur.

— Le poids $\mathsf{poids}(s)$ du sommet $s$ est $(u_{n \times M + s} \mod p)$.

Un exemple de graphe est donné Figure 1.




*Légende :* Un sommet $s$ de poids $p$ est représenté par s (p) .

FIGURE 1 – Le graphe $G(6, 3, 3)$ pour $\widetilde{u_0}$.

On appelle degré sortant d'un sommet $s$ le nombre de successeurs de $s$, c'est à dire $|\mathsf{succ}_{\mathcal{G}}(s)|$.

**Question 3** *Implémenter une fonction calculant le graphe $G(\cdot, \cdot, \cdot)$, et l'utiliser pour calculer la somme $\sum_{s \in \mathcal{N}} |\mathsf{succ}_{\mathcal{G}}(s)|$ des degrés sortants des sommets des graphes $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$ suivants :*

***a)*** $G(6, 3, 10)$ ***b)*** $G(123, 7, 10)$ ***c)*** $G(1\,234, 10, 10)$ ***d)*** $G(10\,001, 22, 10)$

**Question à développer pendant l'oral 2** *Décrire la structure de données que vous avez choisie pour représenter les graphes.*
*On exprimera l'ordre de grandeur de l'espace mémoire utilisé par votre représentation en fonction du nombre de sommets $n$ et du nombre d'arcs $m$ des graphes.*

À partir de maintenant, on ne comptera pas le temps de calcul des graphes, ni leur espace mémoire, dans les évaluations de complexités d'algorithmes.

**Hachage d'un graphe** On définit une fonction de hachage sur les graphes comme suit : pour tout graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$,

$$\mathsf{hash}(\mathcal{G}) = \Big(\sum_{s\in\mathcal{N}} \sum_{t\in\mathsf{succ}_\mathcal{G}(s)} (s+1)\times(t+\mathsf{poids}(t)^2)\Big) \mod 1000.$$

**Question 4** *Calculer la valeur de $\mathsf{hash}(\mathcal{G})$ pour les graphes $\mathcal{G}$ suivants :*

***a)*** $G(6, 3, 10)$ ***b)*** $G(123, 7, 10)$ ***c)*** $G(1\,234, 10, 10)$ ***d)*** $G(10\,001, 22, 10)$

# 2 Marches sur les graphes

Une stratégie sans mémoire $\pi$ pour un graphe $(\mathcal{N}, \mathcal{E}, \mathsf{poids})$ est une fonction associant à chaque sommet $s$ l'un de ses successeurs :

$$\pi : \mathcal{N} \to \mathcal{N} \quad \text{telle que} \quad \forall s \in \mathcal{N},\ (s, \pi(s)) \in \mathcal{E}.$$

Notez qu'il n'existe pas de telle fonction si le graphe possède un sommet sans successeur. Comme les sommets des graphes de ce sujet ont tous au moins un successeur, cela ne sera pas un problème.

Étant donné un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$ avec $\mathcal{N} = \{0, \ldots, n-1\}$, on note $\pi_0(\mathcal{G})$ la stratégie sans mémoire choisissant dans chaque sommet son plus petit successeur :

$$\forall s \in \mathcal{N},\ \ \pi_0(\mathcal{G})(s) = \min\{t \in \mathcal{N} \mid (s,t) \in \mathcal{E}\}.$$

## 2.1 Marches simples

Étant donné un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$, une marche (ou chemin) $\rho$ sur $\mathcal{G}$ est une liste de sommets $s_0, \ldots, s_L$ reliés par des arcs, c'est à dire tel que $(s_i, s_{i+1}) \in \mathcal{E}$ pour tout $0 \le i < L$. On appelle $L$ la longueur de la marche $\rho$ : il s'agit du nombre d'arcs dans $\rho$.

Étant donné un sommet initial $s$ et un entier $L$, une stratégie $\pi$ définit une marche de longueur $L$ en partant du sommet $s$ et en se déplaçant dans le graphe selon la stratégie $\pi$. Plus précisément, on note $\mathsf{marche}(\mathcal{G}, L, \pi, s)$ la marche $s_0, \ldots, s_L$ telle que $s_0 = s$ et $s_{i+1} = \pi(s_i)$ pour tout $0 \le i < L$.

**Question 5** *Implémenter une fonction calculant la marche $\mathsf{marche}(\cdot,\cdot,\cdot,\cdot)$.*
*Pour un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$ où $\mathcal{N} = \{0, \ldots, n-1\}$, évaluer la somme modulo 1000 des sommets de la marche de longueur $L$ au départ du sommet $n-1$ dans $\mathcal{G}$ suivant la stratégie $\pi_0(\mathcal{G})$. C'est-à-dire que si $\mathsf{marche}(\mathcal{G}, L, \pi_0(\mathcal{G}), n-1) = s_0, \ldots, s_L$, alors on cherche la quantité :*

$$\left(\textstyle\sum_{0\le i\le L} s_i\right) \ \mathit{mod}\ 1000$$

*pour les graphes $\mathcal{G}$ et longueurs $L$ suivants :*

**a)** $G(6, 3, 10), L = 4$

**b)** $G(123, 7, 10), L = 20$

**c)** $G(1\,234, 10, 10), L = 100$

**d)** $G(10\,001, 22, 10), L = 5\,000$

**Question à développer pendant l'oral 3** *Donner la complexité en temps et en mémoire de votre algorithme, en fonction du nombre de sommets $n$, du nombre d'arcs $m$, et de la longueur de la marche $L$.*

## 2.2 Marches de concert avec seuil

Cette section est indépendante de la section 3.

Soit un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$ et une stratégie $\pi$ de $\mathcal{G}$. Dans cette section, nous allons étudier des marches de concert dans le graphe : à chaque instant, un certain nombre d'agents seront répartis sur les sommets du graphe, et ceux-ci se déplaceront de concert et simultanément sur le graphe en suivant la même stratégie. De plus, ces déplacements seront soumis à un effet de seuil : pour qu'un groupe d'agents présents dans un sommet se déplace, il sera nécessaire qu'ils soient strictement plus nombreux que le poids du sommet.

Plus précisément, on considère un ensemble fini d'agents $\mathcal{A} = \{0, \ldots, n-1\}$ (il y a donc autant d'agents que de sommets). Ces agents sont répartis sur les sommets du graphe $\mathcal{G}$ : il est possible qu'il y ait plusieurs agents sur le même sommet, et qu'il y ait des sommets sans agents. Initialement, chaque sommet du graphe contient exactement un agent : l'agent $i$ est dans le sommet $i$. Tous les agents se déplacent sur le graphe en suivant la stratégie $\pi_0(\mathcal{G})$. Les agents essaient de se déplacer par groupe (tous les agents dans le sommet $s$ se déplacent ensemble vers le sommet $\pi_0(\mathcal{G})(s)$), et les déplacements sont simultanés (tous les groupes d'agents se déplacent en même temps). Cependant, ces déplacements ne peuvent pas toujours avoir lieu : pour qu'un groupe d'agents $a_1, \ldots, a_k$ dans un sommet $s$ puisse se déplacer en $\pi_0(\mathcal{G})(s)$, il est nécessaire qu'ils soient strictement plus nombreux que le poids du sommet $s$, c'est à dire que $k > \mathsf{poids}(s)$ ; si $k \leq \mathsf{poids}(s)$, les agents $a_1, \ldots, a_k$ restent dans le sommet $s$, en attendant que suffisamment d'autres agents les rejoignent. Une étape de déplacement de *tous* les groupes d'agents le pouvant est appelé un pas, et une séquence de pas décrit une marche de concert.
On souhaite effectuer une marche de concert pour un grand nombre de pas, et sur des graphes relativement importants. Pour être plus efficace, nous utiliserons la notion de sommet vivant : un sommet vivant est un sommet dans lequel il y a au moins un agent, et plus d'agents que le poids du sommet. Autrement dit, les sommets vivants sont ceux depuis lesquels les groupes d'agents vont se déplacer au prochain pas. La Figure 2 décrit un pas d'une marche de concert sur le graphe $G(6, 3, 3)$, ainsi que les sommets vivants lors de cette marche. Par exemple, il y a initialement deux sommets vivants, et il ne reste qu'un sommet vivant après deux pas.
Nous proposons de calculer la marche de concert en maintenant deux structures de données : la première contenant l'état du graphe, c'est-à-dire les agents présents dans chaque sommet ; et la seconde contenant l'ensemble des sommets vivants.

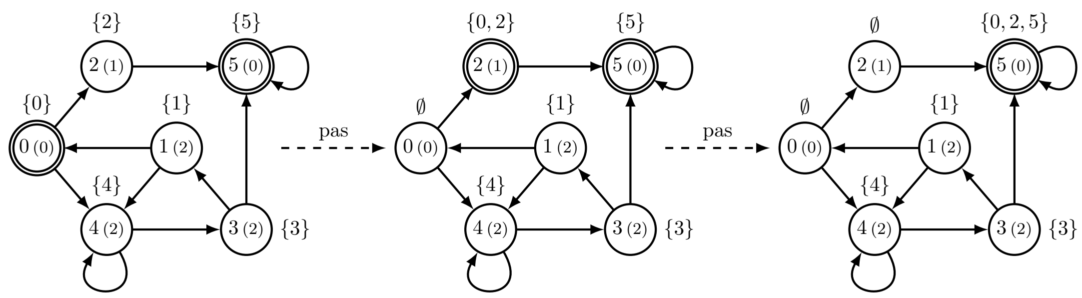
**Question 6** *Implémenter l'algorithme réalisant une marche de concert à l'aide de l'approche proposée ci-dessus. Utiliser cette implémentation pour calculer le nombre de sommets vivants après $L$ pas de la marche de concert dans les graphes suivants :*

**a)** $G(6, 3, 3), L = 4$

**b)** $G(1\,234, 10, 3), L = 100$

**c)** $G(10\,001, 22, 10), L = 50\,000$

**d)** $G(100\,001, 40, 10), L = 100\,000$




*Légende :* Les agents présents dans un sommet sont indiqués au dessus ou à côté de celui-ci. Les sommets vivants sont marqués d'un double cercle. Par exemple, avant le premier pas (donc dans le graphe le plus à gauche), les sommets 0 et 5 sont vivants.

FIGURE 2 – Deux pas d'une marche de concert dans le graphe $G(6, 3, 3)$ pour $\widetilde{u_0}$.

**Question 7** *On note $C(\mathcal{G}, L, s)$ l'ensemble des agents dans le sommet $s$ après $L$ pas de la marche de concert dans le graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \textsf{poids})$. Calculer la quantité :*

$$\Big(\sum_{s\in\mathcal{N}} \sum_{a\in C(\mathcal{G},L,s)} s \cdot a\Big) \ mod \ 1000$$

*pour les graphes et nombres de pas suivants :*

- ***a)*** $G(6, 3, 3), L = 4$
- ***b)*** $G(1\,234, 10, 3), L = 100$
- ***c)*** $G(10\,001, 22, 10), L = 50\,000$
- ***d)*** $G(100\,001, 40, 10), L = 100\,000$

Dans les deux questions à préparer à l'oral suivantes, on exprimera les coûts en temps et en mémoire en fonction, entre autres, du nombre de groupes d'agents et du nombre de groupes d'agents vivant au $i$-ème pas (pour $i$ allant de 0 à $L$).
*Remarque :* on ne demande pas de trouver une formule caractérisant le nombre de groupes d'agents et de groupes d'agents vivants après $i$ pas.

**Question à développer pendant l'oral 4** *Décrire la structure de données que vous avez utilisée pour : i) stocker l'état du graphe ; et ii) stocker l'ensemble des sommets vivants.*
*Donner leur espace mémoire, et le coût en temps pour mettre à jour ces deux structures de données lors de l'exécution d'un pas de la marche de concert.*

**Question à développer pendant l'oral 5** *Donner la complexité en temps et mémoire de l'évaluation de $L$ pas de la marche de concert.*

# 3 Marche et stratégie optimale

Dans cette section, nous allons associer à chaque marche une valeur (entière), puis nous chercherons à déterminer la stratégie optimale à horizon fini $L$, c'est-à-dire la stratégie dont la marche associée (en partant du sommet 0) est de valeur maximale parmi toutes celles de longueur $L$.

### 3.1 Valeur d'une marche

On considère une notion de valeur d'une marche paramétrée par un entier $\alpha \in \mathbb{N}$. Étant donné une marche $\rho$ sur un graphe $\mathcal{G}$, la valeur de $\rho$ pour le paramètre $\alpha$, que l'on note $\mathsf{valeur}_\alpha(\rho)$, est la somme du poids des sommets par lesquels passe $\rho$, en ne comptant pas plus de $\alpha$ fois le poids de chaque sommet. Plus précisément :

$$\mathsf{valeur}_\alpha(\rho) = \sum_{s \in \mathcal{N}} \mathsf{poids}(s) \cdot \min(\alpha, \mathsf{count}(s, \rho))$$

où $\mathsf{count}(s, \rho)$ compte le nombre de passage de $\rho$ dans un sommet $s$.

**Question 8** *Implémenter la fonction $\mathsf{valeur}_\alpha(\cdot)$, et l'utiliser pour calculer*

$$\mathsf{valeur}_\alpha(\rho) \mod 999$$

*pour la marche $\rho$ de longueur $L$ partant du sommet 0 et suivant la stratégie $\pi_0(\mathcal{G})$, pour les graphes $\mathcal{G}$, paramètres $\alpha$ et longueurs de marche $L$ suivants :*

- ***a)*** $G(6, 3, 10), \alpha = 2, L = 6$
- ***b)*** $G(123, 7, 10), \alpha = 20, L = 100$
- ***c)*** $G(1\,234, 10, 10), \alpha = 1000, L = 100\,000$
- ***d)*** $G(10\,001, 22, 10), \alpha = 10\,000, L = 10\,000\,000$
- ***e)*** $G(100\,001, 40, 10), \alpha = 100, L = 1\,000\,000\,000\,000$
- ***f)*** $G(200\,002, 50, 10), \alpha = 100, L = 1\,000\,000\,000\,000$

**Question à développer pendant l'oral 6** *Détailler votre algorithme et donner sa complexité en temps et en mémoire.*

### 3.2 Stratégie avec mémoire optimale

Nous allons maintenant chercher à déterminer la stratégie optimale parmi l'ensemble des marches d'une certaine longueur $L$.
Pour l'instant, nous n'avons considéré que des stratégies sans mémoire. Dans cette section, nous allons nous intéresser à des stratégies plus complexes, les stratégies avec mémoire. Une stratégie avec mémoire $\pi_{\mathsf{h}}$ pour un graphe $(\mathcal{N}, \mathcal{E}, \mathsf{poids})$ est une fonction qui choisit dans quel sommet se déplacer en fonction de l'historique des sommets déjà empruntés. Plus précisément, une stratégie avec mémoire $\pi_{\mathsf{h}}$ est une fonction associant à toute marche non-vide $\rho = s_0, \ldots, s_l$ de $\mathcal{G}$ un successeur de $s_l$ :

$$(s_l, \pi_{\mathsf{h}}(s_0, \ldots, s_l)) \in \mathcal{E} \quad \text{pour toute marche } s_0, \ldots, s_l \text{ de } \mathcal{G}$$

On note $\mathsf{marche}_{\mathsf{h}}(\mathcal{G}, L, \pi_{\mathsf{h}}, s)$ la marche de longueur $L$ partant d'un sommet $s$ et suivant la stratégie avec mémoire $\pi_{\mathsf{h}}$ dans le graphe $\mathcal{G}$. Un exemple de stratégie avec mémoire et de marche est donné Figure 4.
Une stratégie $\pi_{\mathsf{h}}^o$ est optimale pour $\mathsf{valeur}_\alpha(\cdot)$ à horizon fini $L \in \mathbb{N}$ si la marche associée est de valeur maximale parmi toutes les marches de longueur $L$ au départ du sommet 0, c'est-à-dire :

$$\pi_{\mathsf{h}}^o = \underset{\pi_{\mathsf{h}}}{\operatorname{argmax}} \Big(\mathsf{valeur}_\alpha(\mathsf{marche}_{\mathsf{h}}(\mathcal{G}, L, \pi_{\mathsf{h}}, 0))\Big).$$

**Question à développer pendant l'oral 7** *Donner un graphe $\mathcal{G}$ et un horizon $L \in \mathbb{N}$ tels qu'il existe une stratégie avec mémoire sur $\mathcal{G}$ de valeur strictement meilleure que toute stratégie sans mémoire sur $\mathcal{G}$ (à horizon $L$, au départ de $0$).*

**Question 9** *En énumérant toutes les marches de longueur $L$, calculer la valeur selon $\mathsf{valeur}_\alpha(\cdot)$ de la stratégie avec mémoire optimale au départ du sommet $0$ pour les graphes $\mathcal{G}$, horizons $L$ et paramètres $\alpha$ suivants :*

***a)*** $G(6, 3, 10), \alpha = 1, L = 6$

***b)*** $G(6, 3, 10), \alpha = 6, L = 6$

***c)*** $G(1\,234, 10, 10), \alpha = 2, L = 9$

***d)*** $G(10\,001, 22, 10), \alpha = 2, L = 4$

**Question à développer pendant l'oral 8** *Évaluer la complexité en temps et en mémoire de l'exécution de votre algorithme.*

## 3.3 Stratégie avec mémoire optimale : cas $\alpha > L$

Dans le cas où $\alpha > L$, la valeur $\mathsf{valeur}_\alpha(\rho)$ d'une marche $\rho = s_0, \ldots, s_L$ est la somme des poids des sommets de $\rho$, et peut être écrite :

$$\mathsf{valeur}_\alpha(s_0, \ldots, s_L) = \sum_{0 \le i \le L} \mathsf{poids}(s_i)$$

Puisque la valeur précise de $\alpha$ n'a plus d'importance quand $\alpha > L$, on pose $\mathsf{valeur}(\rho) = \mathsf{valeur}_{L+1}(\rho)$ où $L$ est la longueur de $\rho$.

**Question 10** *Calculer la valeur selon $\mathsf{valeur}(\cdot)$ de la stratégie avec mémoire optimale au départ du sommet $0$ pour les graphes $\mathcal{G}$ et horizons $L$ suivants :*

***a)*** $G(6, 3, 10), L = 6$

***b)*** $G(6, 3, 10), L = 200$

***c)*** $G(1\,234, 10, 10), L = 40$

***d)*** $G(10\,001, 22, 10), L = 20$

**Question à développer pendant l'oral 9** *Détailler votre algorithme et donner sa complexité en temps et en mémoire.*

## 3.4 Stratégie optimale pour des marches infinies

Nous considérons maintenant des marches infinies sur un graphe $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathsf{poids})$. Une marche infinie est une liste de sommets $(s_i)_{i \in \mathbb{N}}$ reliés par des arcs, c'est-à-dire telle que $(s_i, s_{i+1}) \in \mathcal{E}$ pour tout $i \in \mathbb{N}$. Plus précisément, nous allons nous intéresser aux marches ultimement cycliques. Une marche infinie est ultimement cyclique si elle est de la forme :

$$\underbrace{s_0, \ldots, s_L}_{\text{préfixe fini}}, \underbrace{t_0, \ldots, t_K}_{\text{cycle}}, \underbrace{t_0, \ldots, t_K}_{\text{cycle}}, \ldots$$

c'est-à-dire si la marche commence par un *préfixe fini* $s_0, \ldots, s_L$ et se termine par un *cycle* $t_0, \ldots, t_K$ se répétant un nombre infini de fois.

La valeur d'une telle marche infinie $\mathsf{valeur}((s_i)_{i\in\mathbb{N}})$ ne peut pas être la somme des poids des sommets $s_i$, puisque cette somme risque d'être divergente. À la place, nous considérerons que la valeur d'une marche infinie ultimement cyclique est la valeur moyenne des poids des sommets du cycle final, c'est-à-dire que si la marche $(s_i)_{i\in\mathbb{N}}$ finit par répéter le cycle $t_0, \ldots, t_K$, alors

$$\mathsf{valeur}_\infty((s_i)_{i\in\mathbb{N}}) = \frac{\sum_{0\leq i\leq K} \mathsf{poids}(t_i)}{K+1}$$

Noter que cette valeur est bien défini, car elle ne dépend pas du cycle choisi.

On pose $H(n, M, p)$ le graphe obtenu à partir de $G(n, M, p)$ en remplaçant tout arc d'un sommet $i$ vers lui-même par un arc de $i$ vers $(i + 1 \mod n)$. Ces graphes sont sans boucles. Un exemple de tel graphe est donné Figure 7.

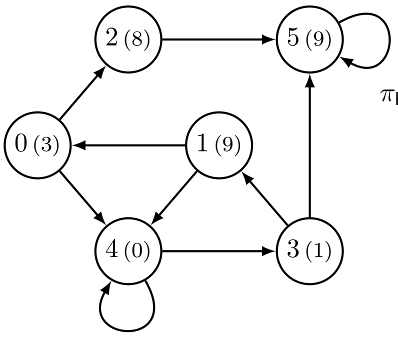
Dans la question suivante, on donnera la valeur d'une marche infinie avec 2 chiffres après la virgule.

**Question 11** *Calculer la valeur selon $\mathsf{valeur}_\infty(\cdot)$ de la marche infinie ultimement cyclique (au départ de n'importe quel sommet) de valeur maximale dans les graphes $\mathcal{G}$ (sans boucles) suivants :*

***a)*** $H(6, 3, 10)$ ***b)*** $H(10, 3, 10)$ ***c)*** $H(40, 4, 10)$ ***d)*** $H(99, 7, 10)$

**Question à développer pendant l'oral 10** *Détailler votre algorithme et donner sa complexité en temps et en mémoire.*

FIGURE 3 – Code OCaml correction question 8.




$\pi_{\mathsf{h}}(s_0, \ldots, s_l, 0) = 4$ $\quad$ $\pi_{\mathsf{h}}(s_0, \ldots, s_l, 1) = 4$

$\pi_{\mathsf{h}}(s_0, \ldots, s_l, 2) = 5$ $\quad$ $\pi_{\mathsf{h}}(s_0, \ldots, s_l, 3) = 1$ $\quad$ $\pi_{\mathsf{h}}(4) = 4$

$$\pi_{\mathsf{h}}(s_0, \ldots, s_l, 4) = \begin{cases} 4 & \text{si } s_{l-1} \neq 4 \text{ (avec } l > 0) \\ 3 & \text{sinon} \end{cases}$$

$$\pi_{\mathsf{h}}(s_0, \ldots, s_l, 5) = 5$$

$$\mathsf{marche}_{\mathsf{h}}(\mathcal{G}, 7, \pi_{\mathsf{h}}, 0) = (0, 4, 4, 3, 1, 4, 4, 3)$$

À droite, une stratégie avec mémoire $\pi_{\mathsf{h}}$ restant dans le sommet 4 deux fois de suite, et la marche de longueur 7 suivant $\pi_{\mathsf{h}}$ et partant de 0.

FIGURE 4 – Exemple de stratégie avec mémoire et de marche sur le graphe $G(6, 3, 10)$ pour $\widetilde{u_0}$.

FIGURE 5 – Code OCaml correction question 9.

FIGURE 6 – Code OCaml correction question 10.

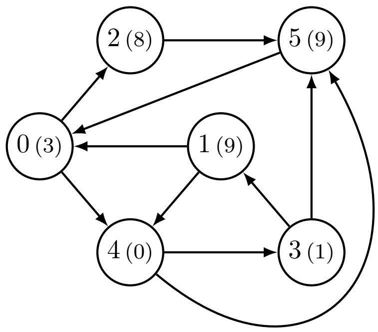



FIGURE 7 – Le graphe sans boucles $H(6, 3, 10)$ pour $\widetilde{u_0}$.

FIGURE 8 – Code OCaml correction question 11.

## Fiche réponse type : Des marches dans des graphes

$\widetilde{u_0}$ : 104

**Question 1**

a) 464

b) 281

c) 919

d) 390

**Question 2**

a) 5

b) 12

c) 20

d) 20

**Question 3**

a) 10

b) 316

c) 3583

d) 36938

**Question 4**

a) 582

b) 629

c) 820

d) 92

**Question 5**

a) 25

b) 284

c) 382

d) 704

**Question 6**

a) 1

b) 63

c) 8

d) 57

**Question 7**

| | |
|---|---|
| a) | 61 |
| b) | 729 |
| c) | 651 |
| d) | 453 |

**Question 8**

| | |
|---|---|
| a) | 29 |
| b) | 108 |
| c) | 310 |
| d) | 41 |
| e) | 313 |
| f) | 461 |

**Question 9**

| | |
|---|---|
| a) | 30 |
| b) | 56 |
| c) | 78 |
| d) | 37 |

**Question 10**

| | |
|---|---|
| a) | 56 |
| b) | 1802 |
| c) | 363 |
| d) | 169 |

**Question 11**

| | |
|---|---|
| a) | 6.67 |
| b) | 4.75 |
| c) | 7.00 |
| d) | 7.57 |

# Fiche réponse : Des marches dans des graphes

**Nom, prénom, $u_0$ :** ..........................................

**Question 1**

a)

b)

c)

d)

**Question 2**

a)

b)

c)

d)

**Question 3**

a)

b)

c)

d)

**Question 4**

a)

b)

c)

d)

**Question 5**

a)

b)

c)

d)

**Question 6**

a)

b)

c)

d)

**Question 7**

a)

b)

c)

d)

**Question 8**

a)

b)

c)

d)

e)

f)

**Question 9**

a)

b)

c)

d)

**Question 10**

a)

b)

c)

d)

**Question 11**

a)

b)

c)

d)