Éviter les carrés abéliens

Étant donné un mot w sur un alphabet $\Sigma = Z/kZ$, on note $|w|$ la longueur et $|w|_x$ le nombre d'occurrences d'une lettre $x \in \Sigma$ dans le mot w. Deux mots w et w' sont des *jumeaux abéliens*, ce que l'on note $w \equiv w'$, si $|w|_x = |w'|_x$ pour toute lettre $x \in \Sigma$. Enfin, un mot w est un *carré abélien* s'il admet une factorisation $w = uv$ telle que $u \equiv v$, et w *évite* les carrés abéliens si aucun de ses facteurs non vides n'est un carré abélien.

Question 1. Parmi les mots ci-dessous, lesquels sont des carrés abéliens ? Lesquels évitent les carrés abéliens ?

102120 012330 0123012 102040203040301

Question 2. Démontrer que nul mot de longueur 8 sur l'alphabet Z/3Z n'évite les carrés abéliens.

Ci-dessous, on souhaite démontrer qu'il existe des mots arbitrairement longs, sur l'alphabet $\Sigma = Z/5Z$, qui évitent les carrés abéliens. Dans ce but, on note **W** le dernier des quatre mots donnés en question 1, puis $\sigma : \Sigma^* \to \Sigma^*$ et $\phi : \Sigma^* \to \Sigma^*$ les morphismes de monoïdes définis par $\sigma : k \mapsto k+1$ et $\phi : k \mapsto \sigma^k(\mathbf{W})$. On suppose alors qu'il existe un mot w, aussi court que possible, évitant les carrés abéliens et tel que $\phi(w)$ n'évite pas les carrés abéliens ; cette supposition se révèlera incorrecte.

Question 3. Comment, à l'aide d'un ordinateur, s'assurer que $|w| \geqslant 3$?

On dit que neuf mots $(k, \ell, m, p, q, r, s, t, u)$ forment un *beau* nonuplet si (1) chacun des mots k, ℓ et m est vide ou réduit à une lettre ; (2) p est un suffixe de $\phi(k)$, rs est une factorisation de $\phi(\ell)$, u est un préfixe de $\phi(m)$; (3) chacun des mots p, r, s et u est de longueur au plus 14 ; et (4) $p\phi(q)r \equiv s\phi(t)u$, tandis que $q \not\equiv \ell t$, $q \not\equiv \ell tm$, $kq \not\equiv \ell t$, $kq \not\equiv \ell tm$, $q\ell \not\equiv t$, $q\ell \not\equiv tm$, $kq\ell \not\equiv t$ et $kq\ell \not\equiv tm$.

Question 4. Pourquoi existe-t-il nécessairement un beau nonuplet $(k, \ell, m, p, q, r, s, t, u)$ tel que $w = kq\ell tm$?

Question 5. Démontrer que $|q| - 1 \leqslant |t| \leqslant |q| + 1$.

Question 6. Démontrer que $|p|_x + 2|q| + 5|q|_x + |r|_x = |s|_x + 2|t| + 5|t|_x + |u|_x$ pour toute lettre $x \in \Sigma$.

Question 7. Comment, à l'aide d'un ordinateur, s'assurer qu'il n'existe aucun beau nonuplet ?

Question 8. On suppose les vérifications des questions 3 et 7 effectuées [1]. Démontrer, pour tout entier $n \geqslant 0$, qu'il existe un mot w de longueur n sur l'alphabet $\Sigma = Z/5Z$ qui évite les carrés abéliens.

1. Cela a pris cinq secondes sur l'ordinateur des examinateurs.

Éléments de correction et attentes de l'examinateur

Question 1. Le premier mot est un carré abélien, donc ne les évite pas.

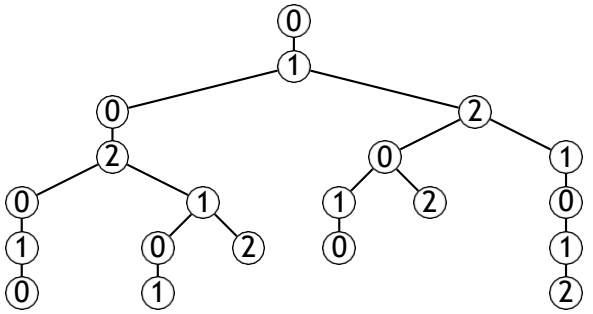
Le second mot n'est pas un carré abélien car il compte un seul 2, mais il a le carré abélien 3 3 pour facteur.

Le troisième n'est pas un carré abélien car il compte un seul 3. En outre, si un de ses facteurs est un carré abélien, ce facteur doit contenir une lettre en deux exemplaires, de part et d'autre du 3, donc il contient le seul 3 du mot et n'est pas abélien malgré tout.

De même, le quatrième mot n'est pas un carré abélien, car il contient onze 0. De surcroît, si l'un de ses facteurs est un carré abélien, intéressons-nous aux lettres autres que 0 dont ce facteur devrait contenir les deux exemplaires. Si une lettre λ est coincée entre ces deux exemplaires de μ, on note cela $\mu \to \lambda$, et alors le facteur devra contenir deux exemplaires de λ. Ici, on a $2 \leftrightarrow 4 \leftrightarrow 3$, et 1 pointe vers toutes ces lettres. Ainsi, le facteur contient des 2 et des 3, mais tous les 2 sont à gauche des 3, donc le facteur n'est pas un carré abélien malgré tout.

Cette question a pour but de permettre au candidat de s'approprier les concepts de l'énoncé, mais également de jauger ses réactions dans un contexte où toute approche basée sur une énumération brutale et exhaustive trouve rapidement ses limites.

Question 2. Essayons de construire un mot qui évite les carrés abéliens. On construit alors un arbre de possibilités, que l'on élague le plus possible : il s'agit de l'arbre des préfixes de l'ensemble des mots évidant les carrés abéliens. Sans perte de généralité, on s'intéresse uniquement aux mots commençant par les lettres 0 1.



On peut en effet vérifier que chaque extension d'un nœud de cet arbre aboutit à un mot dont un suffixe est un carré abélien. Cette tâche est relativement pénible, mais finalement assez rapide, et très simple à mener si l'on est méthodique.

Cette question vise à confirmer que le candidat est capable de se montrer méthodique, ce qui légitimera le fait d'invoquer, dans les questions suivantes, qu'il suffit de se ramener à un nombre fini de cas. L'utilisation d'arbres préfixes n'était pas spécifiquement attendue, même si celle-ci pouvait s'avérer commode.

Il est à noter que, comme la question précédente, cette question pouvait s'avérer chronophage, et l'a été en pratique. Les examinateurs ont bien sûr tenu compte de cet état de fait : être conceptuellement facile ne signifie pas être facile, et ces deux questions sont effectivement assez difficiles.

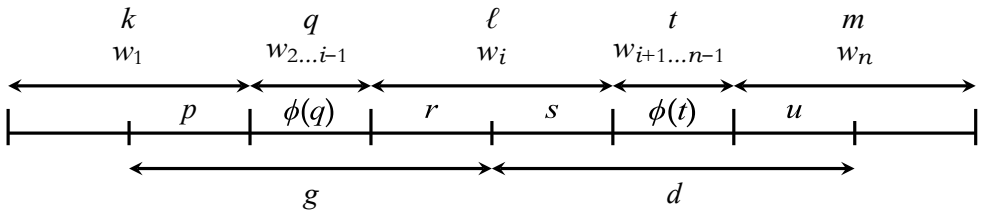
Question 3. Il suffit de vérifier, à l'ordinateur, qu'aucun mot $w \in \Sigma^{\leqslant 2}$ évitant les carrés abéliens n'a une image contenant un carré abélien. Pour ce faire, il suffit donc d'énumérer naïvement les $1 + 5 + 5^2 = 31$ mots concernés, et de disposer d'un algorithme décidant si un mot est un carré abélien. Cette tâche serait affreusement fastidieuse à la main, mais elle est facile à coder.

Cette question vise, par contraste avec la précédente, à s'assurer que le candidat est à même de proposer une approche de nature algorithmique. Ici, effectuer une recherche exhaustive à l'ordinateur n'a rien d'inhumain, au vu du petit nombre de cas à traiter aussi bien que de la simplicité des tests à effectuer. En particulier, il n'était pas attendu du candidat qu'il discute de la complexité de son programme, celle-ci étant manifestement polynomiale (cubique si l'on s'y prend naïvement) en la longueur des mots considérés.

Question 4. Posons $n = |w|$. Il suffit de prendre la factorisation suivante en deux moitiés g et d, qui commence nécessairement en $\phi(w_1)$, s'arrête en $\phi(w_n)$ par minimalité de n, et nécessite éventuellement de couper un mot $\phi(w_i)$ en deux. Nos mots recherchés sont alors ceux représentés ci-dessous.

On a bien $2 \leqslant i \leqslant n-1$, car sinon une moitié g ou ℓ sera comprise dans $\phi(w_1)$ ou $\phi(w_n)$, et l'autre sera plus longue. En outre, pour éviter les cas bâtards où p, r, s ou u serait de longueur $15 = |\mathbf{W}|$, on s'autorise à intégrer k, ℓ ou m à l'un ou l'autre des mots q ou t.

Enfin, le mot $q\ell t$ est de longueur au moins $n-2$, donc n'est pas vide. Par conséquent, si l'une des relations $\equiv$ à éviter était satisfaite, on aurait trouvé un facteur non vide de w qui soit un carré abélien.



Cette question est **difficile**. Son but principal est de mettre en évidence une factorisation de w qui nous permettra ensuite d'aboutir à une absurdité. Elle requiert d'oser faire des dessins et de s'intéresser au rapport que pourraient avoir tous nos mots avec le mot w dont on a supposé l'existence. Une fois ce pas franchi, la question devient d'un seul coup beaucoup plus abordable.

Question 5. Puisque $|p| + 15|q| + |r| = |g| = |d| = |s| + 15|t| + |u|$, on sait que $15(|t| - |q|) = |p| + |r| - |s| - |u|$ est majoré, en valeur absolue, par 2×14, donc que $|t| - |q|$ vaut au plus 1 en valeur absolue.

Cette question sert de préparatif à la question suivante. Elle a également pour but de s'assurer que le candidat est à même de prendre du recul après le travail minutieux que requerrait la question précédente. Ici, il s'agit simplement de comprendre, au moins intuitivement, que $|q|$ et $|t|$ valent à peu près un quinzième de la longueur commune des deux moitiés du carré abélien contenu dans $\phi(w)$, c'est-à-dire g et d.

Question 6. Le mot $\mathbf{W}$ contient deux exemplaires de chaque lettre, ainsi que cinq exemplaires surnuméraires de la lettre 0. En d'autres termes, $|\phi(y)|_x = |\mathbf{W}|_{x-y} = 2 + 5 \times \mathbf{1}_{x=y}$. Par conséquent, $|p|_x + 2|q| + 5|q|_x + |r|_x = |p|_x + |\phi(q)|_x + |r|_x = |g|_x$ et, de même, $|s|_x + 2|t| + 5|t|_x + |u|_x = |d|_x$. Puisque gd est un carré abélien, ces deux quantités sont donc égales.

La principale difficulté est ici de donner un sens aux quantités données dans l'énoncé. L'apparition des termes $|p|_x$ et $|r|_x$ doit faire penser au nombre de x dans un mot comportant les facteurs p et r, et l'on n'a alors plus guère de choix.

Question 7. Si l'on dispose d'un beau nonuplet et que x est une lettre commune à q et à t, supprimer une occurrence de x dans q et dans t nous fournit un autre beau nonuplet. On considère donc un éventuel beau nonuplet pour lequel $|qt|$ est minimal : pour toute lettre $x \in \Sigma$, on a $\min\{|q|_x, |t|_x\} = 0$, de sorte que

$$\pm 5 \max\{|q|_x, |t|_x\} = 5(|q|_x - |t|_x) = |s|_x + |u|_x - |p|_x - |r|_x + 2(|t| - |q|)$$

est majoré, en valeur absolue, par $7 + 7 + 2$, et donc que que $\max\{|q|_x, |t|_x\} \leqslant 2$.

Il existe donc un nombre fini de beau nonuplets potentiels, et il suffit de vérifier qu'aucun d'entre eux n'est effectivement beau.

L'enjeu de cette question **difficile** est de voir comment se ramener à un nombre de cas fini, comme en question 3, c'est-à-dire à ne travailler qu'avec des mots q et t courts. Pour ce faire, vu l'enchaînement des questions, il va manifestement falloir exploiter la mystérieuse question 6 et son résultat : au vu de la question 5, elle nous informe que $|q|_x$ et $|t|_x$ sont forcément assez proches. La difficulté principale consiste donc à s'assurer que ces deux quantités sont petites, et c'est ici qu'intervient l'idée de supprimer chaque lettre x apparaissant à la fois dans q et dans t.

Cette question a pour but de permettre aux tout meilleurs candidats de tirer leur épingle du jeu. Il n'était pas attendu que ceux-ci terminent la question, mais plutôt qu'ils engagent une discussion avec l'examinateur pour proposer des idées à même d'aller dans le bon sens.

Question 8. Les questions 3 à 7 démontrent que la supposition effectuée dans l'encadré n° 2 est incorrecte. En particulier, le morphisme ϕ^k transforme $\mathsf{0}$ en un mot de longueur 15^k qui évite tout carré abélien, et tout facteur de ce mot évite également les carrés abéliens.

Cette question joue le rôle de conclusion pour le problème. En pratique, il n'était pas attendu qu'elle soit abordée.

Nœuds importants d'un arbre

On considère un arbre à n nœuds. L'*excentricité* d'un nœud est sa distance maximale à un autre nœud de l'arbre. Le *diamètre* d'un arbre est le plus long chemin entre deux nœuds de l'arbre. Le *rayon* d'un arbre est la plus petite excentricité d'un nœud.

Question 1. Donner un algorithme pour déterminer l'excentricité d'un nœud, et sa complexité.

Question 2. Montrer qu'il y a toujours au plus deux nœuds d'excentricité minimale.

Question 3. Donner un algorithme qui identifie efficacement un tel nœud d'excentricité minimale.

Question 4. Montrer que le rayon r d'un arbre et le diamètre D d'un arbre vérifient $r = \lceil D/2 \rceil$.

Question 5. Donner un algorithme efficace pour identifier le plus long chemin dans un arbre.

Question 6. On imagine que ces nœuds représentent des ordinateurs connectés entre eux et on doit choisir un unique nœud qu'on va relier à Internet, qui partagera sa connexion avec sa composante connexe. En cas de coupure d'une unique arête de l'arbre, on souhaite minimiser le nombre d'ordinateurs privés d'Internet. Donner une méthode qui identifie le nœud à connecter à Internet dans le pire cas.

Question 7. On suppose maintenant l'arbre pondéré par des poids entiers w_{uv} entre u et v. On s'intéresse à colorier k nœuds de façon que si $d(v)$ est le plus court chemin du nœud v à un nœud colorié, la valeur maximale de d sur le graphe soit aussi petite que possible. On demande de renvoyer cette distance dans le pire cas.

Éléments de correction

Question 1. On fait un parcours en profondeur à partir du nœud choisi u, en $O(n)$. L'excentricité $e(u)$ correspond à la hauteur $h(u)$ dans l'arbre formé par le parcours en profondeur et vérifie

$$e(u) = h(u) = 1 + \max_{v \text{ fils de } v} h(u).$$

Question 2. Supposons qu'il y ait deux nœuds u, v d'excentricité minimale $e(u) = e(v)$. On va montrer qu'ils sont forcément reliés. Supposons par l'absurde qu'il existe un x voisin direct de u sur le chemin de u à v. On considère un plus long chemin partant de u. Si celui-ci ne passe pas par x, $e(v) > e(u)$ ce qui est contradictoire. Donc il passe par x d'excentricité au moins $e(u)$ par minimalité de u ; si à présent on considère un plus long chemin partant de x : s'il ne passait pas par u, on aurait $e(u) > e(x) \geqslant e(u)$. Mais s'il passe par u alors on a encore $e(v) > e(x) \geqslant e(u) = e(v)$. Donc u et v sont reliés et si on avait 3 nœuds d'excentricité minimale, ils seraient reliés deux à deux et on aurait un cycle.

Question 3. Intuitivement, on souhaite partir des feuilles, les nœuds de degré 1, et élaguer au fur et à mesure. Une file de priorité permettrait de faire cela en $O(n \log n)$ mais il existe une solution linéaire. Celle-ci consiste à initialiser une file aux feuilles. Lorsqu'on extrait un nœud, on l'enlève et on met à jour les degrés de ses voisins. Si un voisin devient une feuille ($\deg(v) = 1$), on l'enfile. Le dernier nœud traité fait partie du centre.

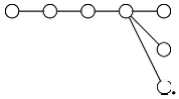
Question 4. Comme dit à la question 1, si on enracine l'arbre en un nœud v, l'excentricité $e(v)$ correspond à la hauteur de l'arbre obtenu. Soit v est sur un plus long chemin, auquel cas le diamètre D est au plus $2e(v)$. Soit v n'est pas sur un plus long chemin, et si on note u le nœud le plus haut d'un plus long chemin et w son extrémité la plus basse, on a que le diamètre D est au plus $2d(u, w) < 2e(v)$. Comme c'est vrai pour tout nœud v, c'est en particulier vrai pour le rayon : $D \leqslant 2r$. Si l'on prend un plus long chemin, s'il est de longueur paire il a un nombre impair de nœuds et celui du milieu a pour excentricité la moitié (si elle était plus grande, on trouverait un chemin encore plus long). Si le plus long chemin est de longueur impaire $2k + 1$, alors un des deux nœuds du milieu est à distance $k + 1$ d'un autre nœud du chemin et son excentricité est $k + 1$. On a donc $r = \lceil D/2 \rceil$.

Question 5. Une méthode qui fonctionne mais difficile à prouver rigoureusement [2] est de faire deux parcours en profondeur. On choisit un nœud a et on identifie le nœud le plus éloigné b avec un DFS. Ensuite, on trouve le point c le plus éloigné de ce point b. Le diamètre est la distance etre b et c.

On peut plutôt le résoudre en un seul parcours par programmation dynamique. On choisit un nœud arbitraire a comme racine de l'arbre puis on s'intéresse à calculer deux quantités, $h(u)$ la hauteur de u i.e. la longueur maximale d'une branche partant de u et $t(u)$ la longueur maximale d'un chemin dans le sous-arbre enraciné en u. La quantité recherchée est $t(a)$ où a est la racine. Si u est une feuille alors on a $h(u) = t(u) = 0$. Sinon on a :

$$h(u) = 1 + \max_{v \text{ fils de } u} h(v)$$

$$t(u) = \begin{cases} \max(h(u), t(v)) & \text{si } u \text{ n'a qu'un fils } v \\ \max\left(\max_{v \text{ fils de } u} t(v),\ 2 + \max_{v_1, v_2 \text{ fils de } u} h(v_1) + h(v_2) \right) & \text{sinon.} \end{cases}$$

Question 6. Il s'agit pour cette question de trouver le *centroïde* de l'arbre, qui n'est pas confondu avec le *centre* de l'arbre (i.e. les nœuds d'excentricité minimale), cf.  . On peut le faire en un seul parcours qui calcule la taille $s(u)$ de chaque sous-arbre. Si u est une feuille alors $s(u) = 1$ qui est aussi le pire cas. Sinon pour chaque fils v de u on calcule le pire cas entre $\max_v s(v)$ (couper l'arête du fils) et $n - s(u)$ (couper l'arête du père).

Question 7. Dichotomie sur la pire distance et on regarde s'il existe un choix des nœuds qui réalise cette distance. C'est un problème de programmation dynamique : on choisit une racine arbitraire, chaque sous-arbre est soit couvert avec le nœud coloré le plus proche à distance d ; soit a des nœuds non couverts, le plus loin étant à distance d.

2. Eindhoven Tuesday Afternoon Club et al. (2002). "On computing a longest path in a tree". In : *Information Processing Letters* 81.2, p. 93-96