ÉCOLE NORMALE SUPÉRIEURE DE LYON

Concours d'admission session 2020

Filière universitaire : Second concours

COMPOSITION D'INFORMATIQUE

Durée : 3 heures

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

⋆ ⋆ ⋆

Ce sujet comporte 3 parties qui ne sont pas indépendantes. Il est toutefois possible d'admettre le résultat de toute question pour répondre aux questions suivantes. Les questions marquées d'une astérisque $(*)$ sont supposées être plus difficiles.

L'énoncé de ce sujet est écrit en utilisant PYTHON comme langage de référence. Cependant, lorsqu'un algorithme est demandé, il peut être écrit en PYTHON, en pseudo-code ou dans un langage au choix du candidat, en utilisant les structures de contrôle habituelles. Toutes les réponses devront être justifiées.

Étant données deux fonctions $f, g : \mathbb{N}^k \to \mathbb{N}$, nous disons que f est en $O(g(n_1, \ldots, n_k))$ lorsqu'il existe des constantes $M, N_0, \ldots, N_k \in \mathbb{N}$ telles que $f(n_1, \ldots, n_k) \leq M \cdot g(n_1, \ldots, n_k)$ pour tous $n_1, \ldots, n_k$ avec $n_i \geq N_i$ pour tout $i = 1, \ldots, k$.

La complexité, ou le temps d'exécution, d'un programme `P` ou d'une séquence d'instructions $o_1, \ldots, o_m$, est le nombre d'opérations élémentaires (addition, multiplication, affectation, test, etc.) nécessaires à l'exécution de `P` (resp. à l'exécution de la séquence d'instructions $o_1, \ldots, o_m$). Cette complexité peut dépendre de paramètres $n_1, \ldots, n_k$ et donc être vue comme une fonction $f : \mathbb{N}^k \to \mathbb{N}$. Dans ce cas nous dirons que `P` (resp. $o_1, \ldots, o_m$) a une complexité en $O(g(n_1, \ldots, n_k))$ lorsque f est en $O(g(n_1, \ldots, n_k))$.

1 Polynômes et idéaux

On désigne par $\mathbb{K}$ l'ensemble des nombres à virgule flottante, muni des opérations usuelles. Un **polynôme à coefficients dans** $\mathbb{K}$ est une fonction de la forme

$$\begin{array}{rccc} f & : & \mathbb{K} & \longrightarrow & \mathbb{K} \\ & & v & \longmapsto & \sum_{i=0}^{n} a_i v^i \end{array}$$

où $a_n, \ldots, a_0 \in \mathbb{K}$ avec $a_n \neq 0$. Un polynôme f comme ci-dessus est représenté par la liste $\ell f = [a_n, \ldots, a_0]$ de ses coefficients. Mathématiquement, on écrit

$$f \quad = \quad \sum_{i=0}^{n} a_i X^i$$

Le **degré** de f est n. Pour $i \in \{0, \ldots, n\}$, on dit que $a_i X^i$ est un **monôme** de f, et le **coefficient** de X^i dans f est a_i. Le **coefficient de poids fort** de f est a_n. Il est utile dans ce qui suit d'adopter la convention selon laquelle le coefficient de X^i dans f est 0 lorsque $i > n$.

On désigne par $\mathbb{K}[X]$ l'ensemble des polynômes à coefficients dans $\mathbb{K}$. L'unique polynôme représenté par la liste vide [ ] est noté $\mathbf{0} \in \mathbb{K}[X]$, avec $\mathbf{0}(v) = 0$ pour tout $v \in \mathbb{K}$. L'ensemble $\mathbb{K}[X]$ est muni des opérations usuelles d'addition $(f, g) \mapsto f+g$ et de multiplication $(f, g) \mapsto f \cdot g$. On a en particulier

$$f + \mathbf{0} = \mathbf{0} + f = f \quad \text{et} \quad f \cdot \mathbf{0} = \mathbf{0} \cdot f = \mathbf{0}$$

La soustraction $(f, g) \mapsto f - g$ peut être définie par $f - g := f + (-1X^0) \cdot g$, et on a $f - f = \mathbf{0}$.

1.1 Algorithmes

Pour les algorithmes impliquant des polynômes, nous adoptons la convention suivante : si ℓf est une liste $[a_n, \ldots, a_0]$ d'éléments de $\mathbb{K}$, alors f est le polynôme $\sum_{i=0}^{n} a_i X^i$.

Question 1. *Donner un algorithme* `somme`$(\ell f, \ell g)$ *qui prend en arguments deux listes* $\ell f = [a_n, \ldots, a_0]$ *et* $\ell g = [b_n, \ldots, b_0]$ *d'éléments de* $\mathbb{K}$*, et qui renvoie la liste* ℓh *des coefficients du polynôme* $h = f + g$*. La complexité de l'algorithme doit être en* $O(\max(n, m))$*.*

Question 2. *Donner un algorithme* `multmon`$(\ell f, m, b)$ *qui prend en arguments une liste* $\ell f = [a_n, \ldots, a_0]$ *d'éléments de* $\mathbb{K}$*, un entier* $m \geq 0$ *et un élément* b *de* $\mathbb{K}$*, et qui renvoie la liste des coefficients du polynôme* $h = (bX^m) \cdot f$*. La complexité de l'algorithme doit être en* $O(n)$*.*

1.2 Idéaux de polynômes

Un ensemble de polynômes $J \subseteq \mathbb{K}[X]$ est un **idéal** si :
- J est non vide, et
- pour tous $f, g \in J$, on a $f + g \in J$, et
- pour tout $f \in J$ et tout $h \in \mathbb{K}[X]$, on a $h \cdot f \in J$.

Question 3. *Soit* $J \subseteq \mathbb{K}[X]$ *un idéal. Montrer que* $\mathbf{0} \in J$*.*

Étant donné un ensemble fini non-vide de polynômes $F = \{f_1, \ldots, f_k\} \subseteq \mathbb{K}[X]$, on pose

$$\langle F \rangle \quad := \quad \{h_1 \cdot f_1 + \cdots + h_k \cdot f_k \mid h_1, \ldots, h_k \in \mathbb{K}[X]\}$$

Par convention, on pose $\langle \emptyset \rangle := \{\mathbf{0}\}$.

Question 4. *Soit* F *un ensemble fini de polynômes. Montrer que* $\langle F \rangle$ *est un idéal.*

Un polynôme $f = \sum_{k=0}^{n} a_i X^i$ est **normalisé** si $a_n = 1$. Notons que $\mathbf{0}$ n'est pas un polynôme normalisé.

Question 5. *Soit* $F \subseteq \mathbb{K}[X]$ *un ensemble fini de polynômes. Montrer qu'il existe un ensemble fini de polynômes normalisés* $\tilde{F} \subseteq \mathbb{K}[X]$ *tel que* $\langle F \rangle = \langle \tilde{F} \rangle$*.*

Congruence induite par un idéal. Soit $J \subseteq \mathbb{K}[X]$ un idéal. Étant donnés $f, g \in \mathbb{K}[X]$, on dit que f **est congru à** g **modulo** J (notation $f \equiv_J g$) si $f - g \in J$. On admet dans toute la suite les propriétés suivantes de la relation $\equiv_J$:

(a) Pour tout $f \in \mathbb{K}[X]$, on a $f \equiv_J f$.

(b) Pour tous $f, g \in \mathbb{K}[X]$ tels que $f \equiv_J g$, on a $g \equiv_J f$.

(c) Pour tous $f, g, h \in \mathbb{K}[X]$ tels que $f \equiv_J g$ et $g \equiv_J h$, on a $f \equiv_J h$.

L'objet de ce sujet est l'étude d'une représentation effective de $\equiv_{\langle F \rangle}$, où $F \subseteq \mathbb{K}[X]$ est un ensemble fini de polynômes. Cette représentation est développée au §3. Elle repose sur une utilisation particulière des graphes dirigés, présentée au §2.

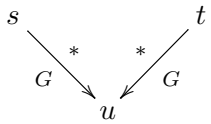
2 Graphes dirigés

Un **graphe dirigé** G est donné par un ensemble S de **sommets** et un ensemble $A \subseteq S \times S$ d'**arrêtes**. Dans ce sujet, nous ne considérons que des graphes dirigés, et dans toute la suite par « graphe » nous entendrons « graphe dirigé ».

Remarque. Un graphe $G = (S, A)$ peut être infini, au sens où les ensembles S et A peuvent être infinis.

Chemins finis. Soit $G = (S, A)$ un graphe. Étant donnés deux sommets (pas nécessairement distincts) $s, t \in S$, un **chemin** de s à t est une suite finie de sommets $s_0, \ldots, s_k \in S$ telle que :
- $s = s_0$, et
- $t = s_k$, et
- $(s_i, s_{i+1}) \in A$ pour tout $i \in \{0, \ldots, k-1\}$.

Notations. Soit $G = (S, A)$ un graphe, et soient $s, t \in S$ deux sommets.
- On note $s \to_G t$ si $(s, t) \in A$.
- On dit que s se **réduit** en t dans G (notation $s \to_G^* t$ ou $s \xrightarrow[G]{*} t$) s'il existe un chemin de s à t dans G.
- On dit que s et t sont **joignables** dans G (notation $s \downarrow_G t$) s'il existe un sommet $u \in S$ tel que $s \to_G^* u$ et $t \to_G^* u$:



- On dit enfin que s et t sont **convertibles** dans G (notation $s \longleftrightarrow_G^* t$) s'il existe une suite finite de sommets $s_0, \ldots, s_k \in S$ telle que
  - $s_0 = s$, et
  - $s_k = t$, et
  - pour tout $i \in \{0, \ldots, k-1\}$, on a soit $s_i \to_G^* s_{i+1}$, soit $s_{i+1} \to_G^* s_i$.

On admet dans toute la suite les propriétés suivantes de la relation $\longleftrightarrow_G^*$:

(i) Pour tout sommet $s \in S$, on a $s \longleftrightarrow_G^* s$.

(ii) Pour tous sommets $s, t \in S$ tels que $s \longleftrightarrow_G^* t$, on a $t \longleftrightarrow_G^* s$.

(iii) Pour tous sommets $s, t, u \in S$ tels que $s \longleftrightarrow_G^* t$ et $t \longleftrightarrow_G^* u$, on a $s \longleftrightarrow_G^* u$.

Question 6. *Soit $G = (S, A)$ un graphe et soient $s, t \in S$ deux sommets tels que $s \downarrow_G t$. Montrer que $s \longleftrightarrow_G^* t$.*

2.1 Confluence et bonne fondation

Soit $G = (S, A)$ un graphe. Par abus de langage, nous dirons qu'un sommet $s \in S$ est une **feuille** s'il n'existe pas de sommet $t \in S$ tel que $s \to_G t$.

Graphes confluents. Un graphe $G = (S, A)$ est **confluent** si on a $s \downarrow_G t$ pour tous sommets $s, t \in S$ tels que $s \longleftrightarrow_G^* t$.

Question 7. *Soit $G = (S, A)$ un graphe confluent. Montrer que pour tout sommet $s \in S$, il existe au plus une feuille t telle que $s \longleftrightarrow_G^* t$.*

Graphes bien fondés et chemins infinis. Soit $G = (S, A)$ un graphe. Un **chemin infini** dans G est une suite infinie de sommets $(s_i)_{i \in \mathbb{N}}$ telle que $s_i \in S$ et $s_i \to_G s_{i+1}$ pour tout $i \in \mathbb{N}$:

$$s_0 \longrightarrow_G s_1 \longrightarrow_G s_2 \longrightarrow_G \cdots \longrightarrow_G s_i \longrightarrow_G s_{i+1} \longrightarrow_G \cdots$$

Un graphe G est **bien fondé** s'il n'existe pas de chemin infini dans G.

Question* 8. *Soit $G = (S, A)$ un graphe bien fondé. Montrer que pour tout sommet $s \in S$, il existe une feuille t telle que $s \to_G^* t$.*

Soit $G = (S, A)$ un graphe confluent et bien fondé et soit $s \in S$ un sommet de G. On notera $\varphi(s)$ l'unique feuille de G telle que $s \longleftrightarrow_G^* \varphi(s)$.

Question 9. *Soit $G = (S, A)$ un graphe confluent et bien fondé. Soient $s, t \in S$ deux sommets de G. Montrer que $s \longleftrightarrow_G^* t$ si et seulement si $\varphi(s) = \varphi(t)$.*

2.2 Algorithmes

Nous allons maintenant voir quelques algorithmes simples sur les graphes que nous utiliserons au §3, et pour lesquels nous supposerons la propriété suivante.

Graphes à branchement fini. Un graphe $G = (S, A)$ est à **branchement fini** si pour tout sommet $s \in S$, l'ensemble

$$\mathrm{Succ}_G(s) \quad := \quad \{t \in S \mid s \to_G t\}$$

des successeurs immédiats de s est fini.

Représentation des graphes à branchement fini. Un graphe $G = (S, A)$ à branchement fini est représenté par un type `graph` et deux fonctions `memb` : `graph` → `bool` et `succ` : `graph` → `list` tels que :

- tous les sommets de G sont des valeurs de type `graph`, et
- une valeur v de type `graph` est un sommet de G si et seulement si `memb`(v) renvoie `True`, et
- si v est un sommet de G alors `succ`(v) renvoie la liste des successeurs immédiats de v (dans un ordre quelconque).

Question 10. *Soit G un graphe à branchement fini représenté par* `graph`*,* `memb` *et* `succ`*. Supposons que G est confluent et bien fondé. Donner un algorithme* `feuille`(s)*, où s est de type* `graph`*, qui renvoie $\varphi(s)$ si s est un sommet de G et qui renvoie* `None` *sinon.*

Question 11. *Soit G un graphe à branchement fini représenté par* `graph`*,* `memb` *et* `succ`*. Supposons que G est confluent et bien fondé. Donner un algorithme* `conv`(s, t)*, où s et t sont de type* `graph`*, qui renvoie* `True` *si s et t sont des sommets de G tels que $s \longleftrightarrow_G^* t$, et qui renvoie* `False` *sinon.*

3 Graphes induits

Soit $f \in \mathbb{K}[X]$ un polynôme normalisé de degré n. Étant donnés deux polynômes $g, h \in \mathbb{K}[X]$, on note $g \to_f h$ lorsqu'il existe un monôme $a \cdot X^m$ de g tel que $m \geq n$ et

$$h \quad = \quad g - aX^{m-n} \cdot f$$

En particulier, on ne peut avoir $\mathbf{0} \to_f h$ car par définition, $\mathbf{0}$ n'a pas de monôme.

Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Le **graphe induit par** F est le graphe $G(F) = (S(F), A(F))$ de sommets $S(F) := \mathbb{K}[X]$ et tel que $(f, g) \in A(F)$ si et seulement s'il existe un $f \in F$ tel que $f \to_f g$.

Nous verrons au §3.2 que l'on a $f \longleftrightarrow_{G(F)}^* g$ si et seulement si $f \equiv_{\langle F \rangle} g$. Nous allons dans un premier temps voir que pour F un ensemble fini de polynômes normalisés, le graphe $G(F)$ est bien fondé. Le §3.3 concerne l'implémentation de la relation $\to_{G(F)}$.

3.1 Bonne fondation du graphe induit

Le **polynôme de poids** d'un polynôme $g = \sum_{k=0}^{n} a_i X^i$ est le polynôme

$$\pi(g) \;:=\; \sum_{k=0}^{n} p_i X^i \qquad \text{où} \qquad p_i \;:=\; \begin{cases} 0 & \text{si } a_i = 0 \\ 1 & \text{sinon} \end{cases}$$

Ainsi, $\pi(g)$ est un polynôme à coefficients entiers. Pour $v \in \mathbb{N}$, nous notons $\pi(g)(v)$ l'entier $\sum_{k=0}^{n} p_i v^i$. Rappelons que pour tout entier $n > 0$, on a

$$2^n \;=\; 1 + \sum_{k=0}^{n-1} 2^k$$

Question 12. *Soit $f \in \mathbb{K}[X]$ un polynôme normalisé et soient $g, h \in \mathbb{K}[X]$ tels que $g \to_f h$. Montrer que $\pi(g)(2) > \pi(h)(2)$.*

Question 13. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Montrer que le graphe $G(F)$ est bien fondé.*

3.2 Graphes induits et congruences

Nous allons maintenant voir que si $F \subseteq \mathbb{K}[X]$ est un ensemble fini de polynômes normalisés, alors pour tous $g, h \in \mathbb{K}[X]$ on a

$$g \longleftrightarrow^*_{G(F)} h \quad \text{si et seulement si} \quad g \equiv_{\langle F \rangle} h$$

Question 14. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Soient $g, h \in \mathbb{K}[X]$ tels que $g \longleftrightarrow^*_{G(F)} h$. Montrer que $g \equiv_{\langle F \rangle} h$.*

Nous allons maintenant montrer l'inverse, c'est-à-dire que $g \equiv_{\langle F \rangle} h$ implique $g \longleftrightarrow^*_{G(F)} h$. Commençons par quelques propriétés de la relation $\to_f$, où $f \in \mathbb{K}[X]$ est un polynôme normalisé.

Question 15. *Soit $f \in \mathbb{K}[X]$ un polynôme normalisé. Montrer que $f \to_f \mathbf{0}$.*

Question 16. *Soient $f, g, h \in \mathbb{K}[X]$ avec f normalisé. Soit $c \in \mathbb{N}$ et $b \in \mathbb{K} \setminus \{0\}$. Montrer que si $g \to_f h$, alors $bX^c g \to_f bX^c h$.*

Question* 17. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés et soient $g, g', h \in \mathbb{K}[X]$. Montrer que si $g \to_{G(F)} g'$, alors $(h + g) \downarrow_{G(F)} (h + g')$.*

Question 18. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Soit $f \in F$, $g \in \mathbb{K}[X]$, $m \in \mathbb{N}$ et $a \in \mathbb{K} \setminus \{0\}$. Montrer que $g \downarrow_{G(F)} g + aX^m f$.*

Question 19. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Soient $f_1, \ldots, f_k \in F$, $m_1, \ldots, m_k \in \mathbb{N}$ et $a_1, \ldots, a_k \in \mathbb{K} \setminus \{0\}$. Montrer que pour tout polynôme $g \in \mathbb{K}[X]$, on a*

$$g \quad \longleftrightarrow^*_{G(F)} \quad g + \sum_{i=1}^{k} a_i X^{m_i} f_i$$

Question 20. *Soit $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. Soient $g, h \in \mathbb{K}[X]$ tels que $g \equiv_{\langle F \rangle} h$. Montrer que $g \longleftrightarrow^*_{G(F)} h$.*

On déduit donc de la Question 14 et de la Question 20, que pour F un ensemble fini de polynômes normalisés, les relations $\equiv_{\langle F \rangle}$ et $\longleftrightarrow^*_{G(F)}$ coïncident. En particulier, comme $G(F)$ est à branchement fini, les techniques du §2 peuvent être utilisées pour décider $\equiv_{\langle F \rangle}$ lorsque $G(F)$ est confluent.

3.3 Algorithmes

Nous allons maintenant étudier l'implémentation de la relation $\to_{G(F)}$, pour $F \subseteq \mathbb{K}[X]$ un ensemble fini de polynômes normalisés. On pourra utiliser les algorithmes du §1.1.

De même qu'au §1.1, si ℓf est une liste $[a_n, \ldots, a_0]$ d'éléments de $\mathbb{K}$, alors f est le polynôme $\sum_{i=0}^{n} a_i X^i$.

Question 21. *Donner un algorithme* `etape`$(\ell g, \ell f, m, a)$ *qui prend en arguments deux listes* $\ell f = [a_n, \ldots, a_0]$ *et* $\ell g = [b_k, \ldots, b_0]$ *d'éléments de* $\mathbb{K}$*, un entier* $m \geq n$ *et un élément* a *de* $\mathbb{K}$*, et qui renvoie la liste des coefficients du polynôme* $h = g - aX^{m-n} \cdot f$*.*

Question 22. *Donner un algorithme* `succsimple`$(\ell g, \ell f)$ *qui prend en arguments deux listes* ℓf *et* ℓg *d'éléments de* $\mathbb{K}$ *et qui renvoie une liste de listes* ℓH *telle que* ℓh *est un élément de* ℓH *si et seulement si* $g \to_f h$*. Les éléments de* ℓH *pourront être dans un ordre quelconque.*

Question 23. *Donner un algorithme* `succ`$(\ell g, \ell F)$ *qui prend en arguments une liste* ℓg *d'éléments de* $\mathbb{K}$ *et une liste de listes* ℓF*, et qui renvoie une liste de listes* ℓH *telle que* ℓh *est un élément de* ℓH *si et seulement si* $g \to_{G(F)} h$*, où* $F := \{f \in \mathbb{K}[X] \mid \ell f \in \ell F\}$*. Les éléments de* ℓH *pourront être dans un ordre quelconque.*

Un ensemble fini de polynômes normalisés $F \subseteq \mathbb{K}[X]$ est une **base de Gröbner** lorsque le graphe induit $G(F)$ est confluent.

Question 24. *Donner un algorithme* `equivideal`$(\ell g, \ell h, \ell F)$ *qui prend en arguments deux listes* ℓg *et* ℓh *d'éléments de* $\mathbb{K}$ *et une liste de listes* ℓF *telle que* $F := \{f \in \mathbb{K}[X] \mid \ell f \in \ell F\}$ *est une base de Gröbner ; qui renvoie* `True` *si* $g \equiv_{\langle F \rangle} h$ *et qui renvoie* `False` *sinon.*

Remarque. Les idéaux de polynômes sont particulièrement intéressants lorsque que l'on considère des polynômes avec un nombre (fixé) fini de variables. L'essentiel de ce sujet se généralise aux polynômes à plusieurs variables, au prix de complications techniques et notationnelles.

Par ailleurs, on peut déterminer effectivement si un ensemble fini de polynômes engendre une base de Gröbner. De plus, l'algorithme de **Buchberger** permet de compléter tout ensemble fini de polynômes en un ensemble fini de polynômes engendrant une base de Gröbner pour le même idéal.

* *
*