Enveloppe Convexe

Samy Jaziri

Sujet de khôlle adapté d'une épreuve pratique d'algorithmique et de programmation du concours commun des écoles normales supérieures : Géométrie (2015).

Préambule

*Tous les algorithmes de ce sujet devront être implémentés en **OCaml**.*

Pour vous mettre dans les conditions du concours, vous n'avez pas le droit d'accéder aux ressources en ligne. Une documentation hors-ligne peut être téléchargée à l'adresse suivante en début d'épreuve : `https://kholles.jaziri.eu/doc-ocaml.zip`

Vous indiquerez vos réponses sur la fiche réponse qui vous a été fournie en utilisant, en entrée de vos programmes, le numéro u_0 inscrit sur cette fiche. Vous remettrez cette fiche à l'examinateur en fin de séance. Une fiche réponse est mise à disposition à la fin du sujet en tant qu'exemple des réponses attendues pour un $\widetilde{u_0}$ particulier.

En ce qui concerne les questions orales de la khôlle, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez pas expliquer votre code ligne par ligne! Quand la complexité d'un algorithme est demandée en temps ou en mémoire en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, donné en notation de Landau $(\mathcal{O}(n), \mathcal{O}(\log(n)), \dots)$. Prenez des notes lorsque vous préparez une question orale pour retrouver plus rapidement les grandes lignes de votre explication lorsque l'examinateur passe vous voir.

Il est recommandé 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.

Il vous est demandé d'aborder les questions dans l'ordre et de noter vos difficultés à répondre à une question avant de passer à la suivante. Vous pourrez alors les aborder avec l'examinateur.

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 Introduction

L'objectif de ce sujet et l'implémenter d'un algorithme de calcul d'enveloppe convexe utilisant la méthode *diviser pour régner*.
On définit dans ce sujet les deux types ci-dessous :

```
type ordre = PLUS_PETIT | PLUS_GRAND | EGAUX ;;
type point = int * int ;;
```

Question 1

Après avoir lu l'intégralité du sujet, définissez un type `ensemble` pour représenter des sous-ensembles de $\mathbb{N}^2$. *Un même point pourra apparaître plusieurs fois dans la représentation de l'ensemble.*

Préparer une réponse à donner à l'oral

Vous devrez motiver le choix des structures de données utilisées.

Tri Rapide Générique

L'algorithme de calcul de l'enveloppe convexe nécessite de trier les points de l'ensemble donné en entrée. On choisi d'utiliser un *tri rapide*. L'ordre utilisé pour la comparaison des points de $\mathbb{N}^2$ sera défini plus tard dans le sujet.

Question 2

Implémenter un algorithme de tri rapide qui prend un paramètre un ensemble et une fonction de comparaison quelconque entre deux éléments de type `point`. La fonction de comparaison aura la signature suivante :

$$\texttt{comparaison} : \texttt{point} \to \texttt{point} \to \texttt{ordre}$$

Préparer une réponse à donner à l'oral

Donnez les grandes lignes de la preuve de correction de l'algorithme de tri rapide.

Génération d'ensembles aléatoires

Considérons $(u_k)_{k\geq 0}$ la suite d'entiers définie par :

$$u_k = \begin{cases} u_0 \text{ (sur votre fiche réponse)} & \text{si } k = 0 \\ 37698 \times u_{k-1} \mod 524287 & \text{si } k > 0 \end{cases}$$

Question 3

Que valent :
a) u_{30} **b)** u_{60} **c)** u_{90}

Pour tout $n \in \mathbb{N}$ on définit l'ensemble

$$E_n = \{(u_{2i}, u_{2i+1}), i \in [\![0, n-1]\!]\}$$

Tri des sous-ensembles de $\mathbb{N}^2$

On définit l'ordre lexicographique sur $\mathbb{N}^2$:

$$\forall (x_1, y_1) \in \mathbb{N}^2, \forall (x_2, y_2) \in \mathbb{N}^2, (x_1, y_1) < (x_2, y_2) \iff x_1 < x_2 \text{ ou } (x_1 = x_2 \text{ et } y_1 < y_2)$$

Définissez une fonction `ordre_lex : point → point → ordre` qui permet de comparer deux points selon l'ordre lexicographique. Puis implémentez une fonction `max_ensemble` qui prend en entrée un sous-ensemble de points de $\mathbb{N}^2$ et renvoie le point le plus grand de l'ensemble pour l'ordre lexicographique.

Question 4

Quels sont les points maximaux de :
a) E_{40} **b)** E_{270} **c)** E_{500}
pour l'ordre lexicographique.

Question 5

Donnez les ensembles et les entiers i suivants, le i^{eme} point de l'ensemble selon l'ordre lexicographique :
a) E_{40}, $i = 21$ **b)** E_{270}, $i = 144$ **c)** E_{500}, $i = 212$

2 Calcul de l'enveloppe convexe d'un sous-ensemble de $\mathbb{N}^2$

Un ensemble $P \subseteq \mathbb{R}^2$ est convexe si pour tous x et y de P, et tout $t \in [0,1], t\,x + (1-t)\,y \in P$. Ce qui peut être formulé ainsi : P est convexe si et seulement si pour tout couple de points de P, le segment reliant ces deux points est inclus dans P.
On appelle enveloppe convexe d'un ensemble E de $\mathbb{R}^2$ l'intersection de tous les ensembles convexes contenant E.

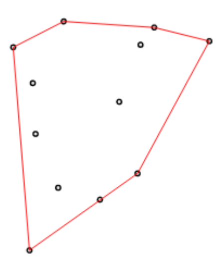

Figure 1: Enveloppe convexe de E_{12}

Étant donné P un ensemble convexe, un point $x \in P$ est dit point extrême de P si l'équation $x = t\,y + (1-t)$ avec $t \in [0,1]$, $y \in P$ et $z \in P$, implique que soit $y = x$, soit $z = x$. On admettra que l'enveloppe convexe d'un ensemble $E \subseteq \mathbb{R}^2$ est égale à l'enveloppe convexe de l'ensemble F des points extrêmes de l'enveloppe convexe de E.

Quand on parlera de déterminer l'enveloppe convexe de E, on parlera en fait de déterminer F, noté $F(E)$, ensemble des points extrêmes de l'enveloppe convexe de E.

Représentation d'une enveloppe convexe

Étant donné trois points $u = (x_u, y_u)$, $v = (x_v, y_v)$ et $w = (x_w, y_w)$ de $\mathbb{R}^2$, on dira que u, v et w sont rangés dans le sens trigonométrique si, intuitivement, lorsque, venant de u et arrivant en v, il faut *tourner à gauche* pour aller vers w.

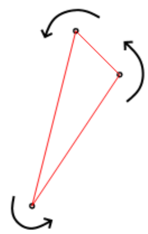

Figure 2: Sens trigonométrique

Formellement u, v et w sont rangés dans le *sens trigonométrique* si et seulement si

$$z_\otimes = (x_u - x_v)\,(y_w - y_v) - (y_u - y_v)\,(x_w - x_v) < 0$$

Si $z_\otimes > 0$ on dira que les points sont rangés dans le *sens anti-trigonométrique*. Enfin si $z_\otimes = 0$ on dira que les points sont *alignés*.
L'enveloppe convexe $F(E)$ d'un ensemble E sera donc représentée par l'*ensemble des points extrêmes de E*, **ordonnés** et tels que :

- le premier point soit le plus petit point de $F(E)$ pour *l'ordre lexicographique*
- trois points consécutifs sont *alignés* ou rangés dans le *sens trigonométrique*
- le dernier point est une copie du premier point, pour des raisons purement algorithmiques.

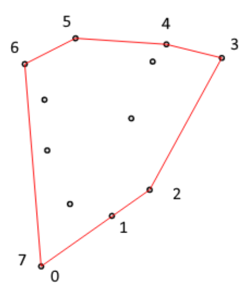

Figure 3: Ordre des éléments dans la représentation de $F(E_{12})$

> *Préparer une réponse à donner à l'oral*
>
> Sachant que l'enveloppe convexe d'un singleton est lui-même. Comment représenter l'enveloppe convexe d'un singleton ?

Principe général

L'algorithme de calcul d'enveloppe convexe par méthode diviser pour régner s'appuie sur un algorithme permettant de calculer l'enveloppe convexe d'un ensemble à partir de l'enveloppe convexe de sa partie gauche et sa partie droite. Le cas de base étant le calcul de l'enveloppe convexe d'un singleton.
Plus précisément, soit deux sous-ensembles E_1, E_2 de $\mathbb{N}^2$ tels que tous les points de E_1 sont plus petit que tous les points de E_2 pour l'ordre lexicographique. On admettra pour l'instant qu'il existe une fonction

$$\texttt{fusion\_enveloppe} : \texttt{ensemble} \rightarrow \texttt{ensemble} \rightarrow \texttt{ensemble}$$

qui peut calculer $F(E_1 \cup E_2)$ en prenant en entrée $F(E_1)$ et $F(E_2)$ en $\mathcal{O}(n_1 + n_2)$ avec n_1 et n_2 les cardinaux de $F(E_1)$ et $F(E_2)$.

Question 6

En utilisant la fonction `fusion_enveloppe`, mais sans l'implémenter, implémentez l'algorithme de calcul de l'enveloppe convexe par la méthode diviser pour régner.

Préparer une réponse à donner à l'oral

Faîte l'analyse de complexité de la l'algorithme de calcul de l'enveloppe convexe que vous avez implémenté.

Recherche de tangentes

Soient deux enveloppes convexes F_1, F_2 telles que tous les points de F_1 soient plus petit que les points de F_2 pour l'ordre lexicographique.
L'algorithme `fusion_enveloppe` sur F_1 et F_2 procède de la sorte consiste à chercher deux paires de points (p_h, q_h) et (p_b, q_b) de $F_1 \times F_2$ tels que $\forall p \in F_1 \cup F_2$:

- q_h, p_h et p sont *alignés* ou rangés dans le *sens trigonométrique*,
- p_b, q_b et p sont *alignés* ou rangés dans le *sens trigonométrique*,

On défini le type suivant :

```
type sens = TRIGO | ANTI_TRIGO | ALIGNE ;;
```

Question 7

Implémentez une fonction `direction` : `point` → `point` → `point` → `sens` qui renvoie le sens dans lequel sont rangés les trois points donnés en entrée.

Pour coder l'algorithme de recherche de tangente entre F_1 et F_2, on procédera de la sorte :

- On initialise p_h et p_b au point maximal de l'ensemble F_1 (à l'aide de la fonction `max_ensemble`).
- On initialise q_b au premier point de l'enveloppe F_2 et q_h au dernier point, qui est une copie du premier.

- **(1)** On fait évoluer p_h dans F_1 en suivant le sens trigonométrique tant que q_h, p_h et le point suivant p_h ne sont pas alignés ou rangé dans le sens trigonométrique.
- **(2)** On fait évoluer q_h dans F_2 en suivant le sens anti-trigonométrique tant que p_h, q_h et le point précédent q_h ne sont pas alignés ou rangé dans le sens anti-trigonométrique.
- On répète **1** puis **2** tant que p_h et q_h peuvent évoluer. On ne fait pas boucler p_h et q_h qui s'arrêtent au pire au dernier point de F_1 et au dernier point de F_2 respectivement.
- **(3)** On fait évoluer p_b dans F_1 en suivant le sens anti-trigonométrique tant que q_b, p_b et le point précédent p_b ne sont pas alignés ou rangé dans le sens trigonométrique.
- **(4)** On fait évoluer q_b dans F_2 en suivant le sens trigonométrique tant que p_b, q_b et le point suivant q_b ne sont pas alignés rangé dans le sens anti-trigonométrique.
- On répète **3** puis **4** tant que p_b et q_b peuvent évoluer. On ne fait pas boucler p_b et q_b qui s'arrêtent au pire au premier point de F_1 et au dernier point de F_2 respectivement.

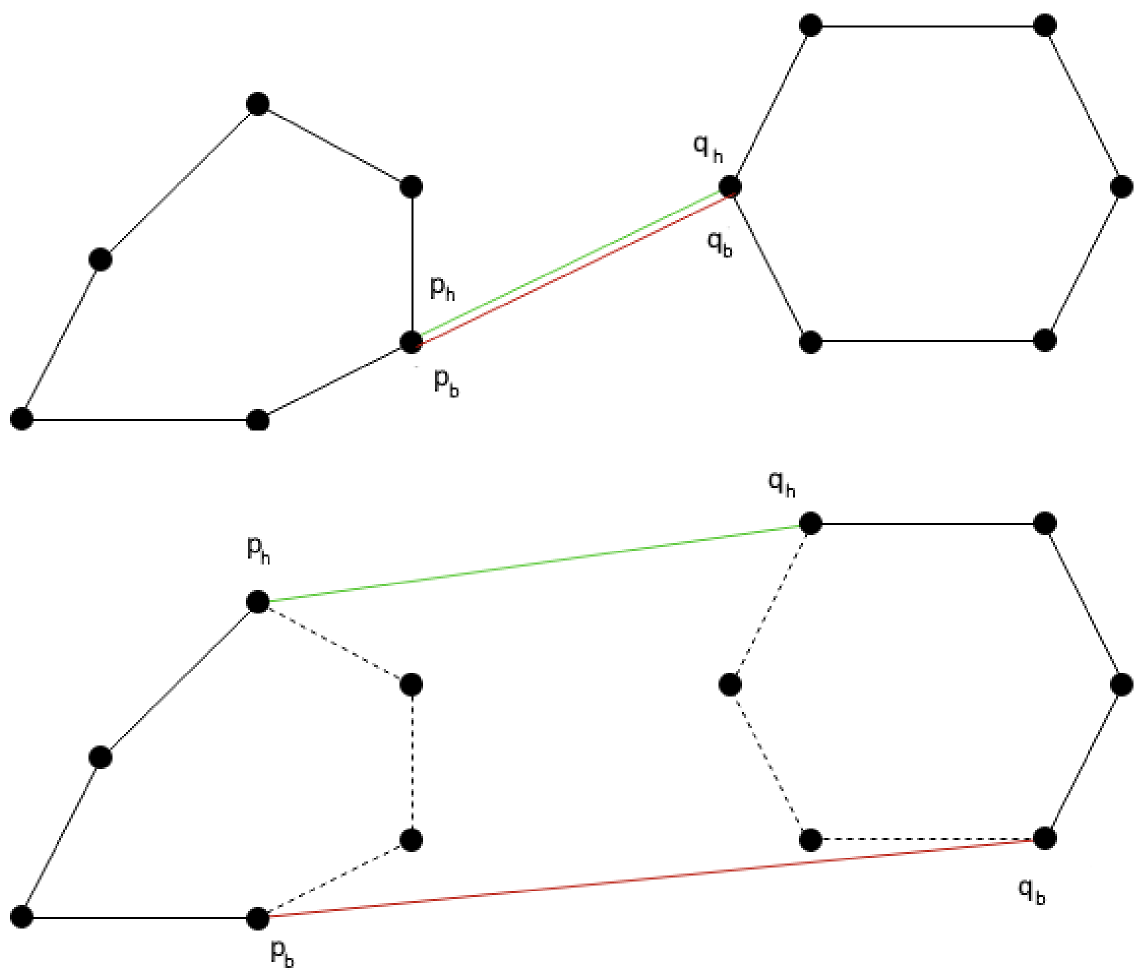

Figure 4: Calcul des tangentes

Question 8

Implémentez deux fonctions `tangente_haute` et `tangente_basse` qui prennent en entrée deux enveloppes convexes F_1, F_2 telles que tous les points de F_1 soient plus petit que les points de F_2 pour l'ordre lexicographique et qui renvoie les couples (p_h, q_h) et (pb, qd)$ respectivement, comme décrit ci-dessus.

L'algorithme de fusion consiste ensuite :

- à calculer les couples (p_h, q_h) et (p_b, q_b) défini ci-dessus
- à éliminer les points de F_1 entre p_b et p_h, exclus.
- à éliminer les points de F_2 avant q_b et après q_h, exclus.
- puis à recoller les points restant en préservant la structure d'enveloppe convexe.

En faisant attention aux cas dégénérés où les enveloppes convexes sont des points ou des droites (singletons et ensemble à une dimension). Implémentez l'algorithme `fusion_enveloppes` utilisé pour le calcul de l'enveloppe convexe.

Question 9

Donnez le cardinal de l'enveloppe convexe des ensembles suivants :
a) E_{100} **b)** E_{212} **c)** E_{550}

sources des images :

- `https://planetcalc.com/8576/`
- `https://algorithmtutor.com/Computational-Geometry/`

Fiche réponse : Enveloppe Convexe

Nom, prénom : DOUZ Nadine

$\widetilde{u_0}$: 12

Question 3:

a) 32026

b) 362800

c) 244169

Question 4:

a) (520605, 132019)

b) (520605, 132019)

c) (520605, 132019)

Question 5:

a) (268212, 181181)

b) (275336, 306789)

c) (244882, 440427)

Question 9:

a) 10

b) 14

c) 18

samy.jaziri@universite-paris-saclay.fr