TP n° 10 – Géométrie

esimon@esimon.fr thomas.h@ens.fr
thomash.fr/teaching/

Dans ce TP, on s'intéressera à l'enveloppe convexe d'un ensemble de points P, c'est à dire au polygone définie par une suite d'éléments de P englobant tous les éléments de P.

Un point sera représenté par une paire `(int * int)`, l'enveloppe convexe par un parcours dans le sens trigonométrique du polygone : $p_0, p_1, \ldots, p_h$.

1 Oups !

Yorel Reivax n'a vraiment aucune idée de l'utilité de ce déterminant :

$$CCW(a,b,c) = \det \begin{pmatrix} 1 & a_x & a_y \\ 1 & b_x & b_y \\ 1 & c_x & c_y \end{pmatrix}$$

Question 1. Écrire la fonction `ccw: int * int -> int * int -> int * int -> int`. Quel lien existe-t-il entre le signe de cette fonction et l'ordre des points [1] ?

2 Parcours de Graham

Question 2. Vérifions d'abord que vous allez faire votre travail correctement. Écrire une fonction `enveloppe: (int * int) list -> (int * int) list -> bool` qui vérifie si un polygone est bien l'enveloppe convexe d'un ensemble de points [2].

Question 3. Écrire une fonction `graham_init: (int * int) list -> int * int` qui étant donné un ensemble de points retourne l'un d'entre eux se trouvant sur l'enveloppe convexe.

Le parcours de Graham est un algorithme qui étant donné un point p_0 sur l'enveloppe convexe, commence par trier les points en fonction de leur angle avec p_0 et l'axe des abscisses ; puis maintient une enveloppe convexe partielle en parcourant les points un à un.

Question 4. Écrire une fonction `graham_it: (int * int) list -> int * int -> (int * int) list` qui étant donné une enveloppe convexe d'au moins 2 points et un nouveau point renvoie une nouvelle enveloppe convexe contenant ce point. Remarquez que l'angle formé par p_0 et ce nouveau point est plus grand que tout autre angle entre p_0 et un point de l'enveloppe convexe partielle.

Question 5. Écrire une fonction `graham: (int * int) list -> (int * int) list` calculant l'enveloppe convexe d'un ensemble de points.

Question 6. Quelle est la complexité de la fonction `graham` [3] ?

Question 7. Prouver que le calcul de l'enveloppe convexe est en $\Omega(n \log(n))$.

Question 8. * Comment s'appelle un polygone à 9 côtés [4] ?

---

1. Vous répondrez à cette question sans savoir que CCW signifie sens antihoraire en anglais.
2. Regardez le CCW d'un point avec n'importe quel segment de l'enveloppe convexe.
3. Si vous répondez $O(2^n)$ ça va barder.
4. Vous veillerez à ne pas faire un disgracieux mélange de racines grecque et latine.

3 Triangulation

Question 9. On appelle triangulation d'un polygone, la décomposition d'un n-gone en $n-2$ triangles. Le poids d'une triangulation est la somme des périmètres des triangles qui la compose.

Écrire une fonction `triangulation_convexe: (int * int) list -> float` qui étant donné un polygone convexe retourne le poids de sa triangulation minimale[5].

La triangulation de Delaunay d'un ensemble P de points du plan est une triangulation $DT(P)$ telle qu'aucun point de P n'est à l'intérieur du cercle circonscrit d'un des triangles de $DT(P)$.

Question 10. Soit un ensemble de points du plan P, et $DT(P)$ sa triangulation de Delaunay, on définie P' la projection des points de P dans l'espace avec $p'_z = |q-p|$ pour q quelconque et p' la projection de p dans P'. À quoi correspond la triangulation de Delaunay $DT(P)$ dans P' ?

Question 11. Il existe un algorithme *diviser pour régner* qui calcule l'enveloppe convexe en deux dimensions en $O(n\log(n))$. Pour recombiner deux enveloppes, il faut ajouter deux de leurs tangentes et retirer les arêtes en trop. Écrire une fonction `dui: (int * int) list -> (int * int) list` qui calcule l'enveloppe convexe en utilisant cette méthode.

Question 12. * Donner une généralisation de cette algorithme en dimension 3.

Question 13. Donner un algorithme qui calcule la triangulation de Delaunay d'un ensemble de points en temps $O(n\log(n))$.

5. Remarquerez qu'une triangulation peut être décomposée en un triangle plus deux triangulations.