- Contrôle continu d'algorithmique géométrique -

Durée : 1h30

Tous documents interdits.
Les exercices sont indépendants et peuvent être traités dans un ordre quelconque.
Le barème est indicatif.
Toutes vos réponses doivent être soignées et, sauf mention contraire, justifiées.

Tous les ensembles de points du plan considérés sont en position générale : ils ne contiennent pas 3 points alignés ni 4 points cocycliques (i.e. sur un même cercle).

- Exercice 1 - Enveloppe convexe, *question de cours/tp* - 3pts -

On s'interesse à l'algorithme de Jarvis (algorithme du 'paquet cadeau').

1. Soient un sommet $P = (x_p, y_p)$ et une droite (A, B) du plan, avec $A = (x_A, y_A)$ et $B = (x_B, y_B)$. Rappeler le calcul exact à effectuer pour décider si P est à droite ou à gauche de (A, B) orientée de A vers B (votre calcul impliquera x_A, y_A, x_B, y_B, x_P et y_P...)
2. Rappeler le principe de l'algorithme de Jarvis ainsi que sa complexité.
3. Dérouler démonstrativement cet algorithme sur un ensemble de 6 points du plan qui contient exactement 4 points sur son enveloppe convexe. Préciser rapidement les différentes étapes.

- Exercice 2 - Intersection de disques - 6pts -

Un disque D_i est donné par son centre $p_i = (x_i, y_i)$ et son rayon r_i. Toutes les valeurs x_i, y_i et r_i sont supposés entières. On souhaite écrire un algorithme $\mathcal{A}$ qui, étant donné un ensemble $\{D_1, \dots, D_n\}$ de n disques dans le plan, détecte si il existe deux indices $i \neq j$ tels que D_i et D_j s'intersectent.

1. Pour deux disques fixés, disons D_1 et D_2, écrire une primitive fonctionnant en temps constant permettant de décider si D_1 et D_2 s'intersectent ou non.
2. Proposer un algorithme fonctionnant en $O(n^2)$ pour répondre au problème posé.
3. On veut améliorer l'algorithme précédent en utilisant une technique par balayage. On a besoin des deux structures de données suivantes :
   - un échéancier T contenant les abscisses triées par ordre croissant des extrémités de chaque disque (le point le plus à gauche, *insertion du disque*, et point le plus à droire, *suppression du disque*). On supposera que toutes ces valeurs sont distinctes.
   - un ordre vertical $\mathcal{O}$ sur les disques, vide au début de l'algorithme.

   Le principe de l'algorithme est alors le même que celui de la détection d'une intersection dans un ensemble de segments.
   (a) Écrire l'algorithme en détail.
   (b) Donner l'exemple d'une famille de disques contenant une seule intersection et tel que celle-ci est détectée à la suppression d'un disque.

(c) En supposant que l'ordre $\mathcal{O}$ soit géré par un arbre de recherche équilibré, permettant l'insertion, la supression, la recherche de prédécesseur et la recherche de successeur en $O(\log n)$, calculer la complexité de votre algorithme.

(d) Prouver la validité de l'algorithme proposé.

- Exercice 3 - Quadrangulation - 6pts -

Un graphe planaire G est une *quadrangulation* si G admet une représentation plane dans laquelle la frontière de chaque face est un cycle de longueur 4.

1. On considère un maillage dans l'espace dont la surface est sans trou et dont chaque face est un quadrilatère. Justifier rapidement que le squelette de ce maillage est une quadrangilation.
2. Dessiner une quadrangulation qui n'est ni un cycle de longueur 4, ni un cube.
3. En utilisant la formule d'Euler ('$n - m + f = 2$') et un double comptage sur la somme des nombres d'arêtes de la frontière de chaque face d'une représentation de G, montrer que $m = 2n - 4$ où n et m désignent respectivement les nombres de sommets et d'arêtes de G.
4. En déduire que G contient un sommet de degré inférieur ou égal à 3.
5. Montrer que si tous les sommets de G ont degré 3 alors G est (le squelette d') un cube.
6. Décrire un encodage de G utilisant $3n \log n$ bits et permettant un test d'adjacence en temps constant.