Examen Géométrie algorithmique.

Maîtrise informatique

avril 2001

1 Formule d'Euler

Le but de cet exercice est de démontrer de manière progressive la formule d'Euler (présentée en cours) :

$$v - e + f = 2$$

pour tout graphe connexe planaire ayant v sommets, e arêtes et f faces.

Définitions/rappels :
Notation : un graphe est noté G. L'ensemble de ses sommets est noté V et l'ensemble de ses arêtes est noté E.

- G est dit *connexe* si, pour tout couple de sommets (u,v) de G, il existe toujours un chemin formé d'arêtes de E permettant de joindre u à v.
- Un *arbre* est un graphe connexe sans cycle.

Le *plongement* d'un graphe dans le plan est (de manière intuitive) une façon de le dessiner en associant à chaque sommet un point du plan et à chaque arête un arc de courbe dans le plan. Un plongement est *planaire* si deux arcs correspondant à des arêtes différentes ne se coupent pas, sauf éventuellement en leur(s) extrémité(s). Un graphe est dit planaire s'il admet au moins un plongement *planaire*.

Question 1- Vérifier la formule sur les cas simples suivants :

Question 2- Soit G un arbre. Montrer que

$$v = e + 1$$

et que la relation d'Euler est donc vérifiée pour un arbre.
Indication : Effectuer une démonstration par récurrence sur le nombre de sommets de G. On pourra utiliser le fait que si G est un arbre ayant au moins deux sommets, alors il a au moins deux sommets de degré 1 (encore appelés *feuilles*).

Question 3- Soit G un graphe connexe planaire avec cycle. Montrer la formule d'Euler pour G.
Indication : Pour cela, on effectuera encore une démonstration par récurrence, mais cette fois sur le nombre d'arêtes de G. Considérer une arête a contenue dans un cycle, l'enlever de G, et examiner les effets induits par cette suppression sur v, e, et f.

Remarque : Le nombre de faces n'est donc pas lié au plongement choisi pour le graphe planaire G.

2 Polygone simple

Étant donné un ensemble P de n points dans le plan. On cherche un polygone simple (les arêtes non-adjacentes ne s'intersectent pas et le polygone n'a pas de trou) qui a exactement les points de P comme sommets.

1. Concevoir un algorithme qui construit un tel polygone en temps $O(n \log n)$.
   (a) Décrire l'algorithme en pseudo-code.
   (b) Analyser sa complexité.
2. Montrer que $\Omega(n \log n)$ est une borne inférieure.

3 k plus proche voisins

Étant donné $\mathcal{S}$ un ensemble de n points dans le plan, on cherche à connaître les k plus proches voisins dans $\mathcal{S}$ d'un point p de $\mathcal{S}$.

-a– Si q_1 est le plus proche voisin de p ($k = 1$) montrer que pq_1 est une arête de la triangulation de Delaunay.

-b– Si q_1 est le plus proche voisin de p et q_2 est le deuxième plus proche voisin de p ($k = 2$) montrer que soit pq_2 soit q_1q_2 est une arête de la triangulation de Delaunay.

-c– Si q_i est le $i$ème plus proche voisin de p (on définit $q_0 = p$) alors montrer que

$\forall i \geq 1 \quad \exists j < i \quad$ tel que $\quad q_iq_j$ est de Delaunay.

-d– Si la triangulation de Delaunay est donnée. Donner en pseudo-code un algorithme de recherche des k plus proches voisins d'un point p.

-e– Pour $k = 1$, quelle est la complexité d'un tel algorithme

-e1– dans le cas le pire pour le choix de p.

-e2– si p est un point choisi au hasard dans $\mathcal{S}$ (tous les points ont une probabilité $\frac{1}{n}$ d'être choisis) .

-f– Mêmes questions pour $k = 2$.