Examen DEA ARAVIS : géométrie algorithmique
14 à 17h, 7 novembre 1997

1 Exercice 1

Algorithme:
Donnée: $\mathcal{S}$ un ensemble de points du plan.
Calculer $\mathcal{V}(\mathcal{S})$ les sommets du diagramme de Vornoï de $\mathcal{S}$
Calculer la triangulation de Delaunay de $\mathcal{S} \cup \mathcal{V}(\mathcal{S})$
Conserver uniquement les arètes de cette triangulation reliant des points de $\mathcal{S}$.
Resultat: des polygones ayant des points de $\mathcal{S}$ comme sommets

- Dessiner le résultat sur l'exemple.
- Un triangle de Delaunay dans la triangulation de $\mathcal{S}$ peut-il faire partie du résultat final. Si oui, a quelle condition, si non, pourquoi ?
- Si $\mathcal{S}$ est constitué de 4 points, dessiner les deux résultats possibles.
- Si $\mathcal{S}$ est constitué des sommets d'un polygone convexe, quel est le résultat
- Si $\mathcal{S}$ est constitué des sommets d'un polygone simple, dessiner un exemple ou le résultat est différent du polygone de départ. Où faut-il rajouter des points pour que le résultat soit le polygone de départ.
- Quel peut être l'objectif d'un tel algorithme

2 Exercice 2

n avions se déplacent en ligne droite à vitesse constante, la position de l'avion i au temps t est $p_i(t) = q_i + tv_i,\ q_i, v_i \in R^2$. Un controleur aérien doit concentrer son attention sur l'avion le plus proche.

- Combien de fois le controleur devra-t-il changer d'avion dans la cas le pire?
- Si il suivait l'avion le plus loin ?
- Si il s'agissait d'objets tombant en chute libre $p_i(t) = q_i + tv_i + t^2 g$ (g est indentique pour tous les objets)

3 Exercice 3

La taille d'une enveloppe inférieure de segments est $\Theta(n\alpha(n))$ et l'algorithme vu en cours calcule cette enveloppe en temps $O(n\alpha(n)\log n)$. Nous allons développer dans cet exercice un algorithme de complexité optimale $\Theta(n \log n)$.

• Étudier le problème du calcul de l'enveloppe inférieure de n segments coupant la droite verticale $x = 0$. (Taille de l'enveloppe, algorithme, temps de calcul).

• Étant donné n segments (quelconques). On note $x_1 < x_2 < \ldots < x_{2n}$ les abcisses des extrémités des segments. On construit un arbre binaire équilibré tel que la racine correspond à l'intervalle $[x_1, x_{2n}]$ et les fils de $[x_i, x_j]$ sont $[x_i, x_k]$ et $[x_k, x_j]$ avec $k = \lfloor (i+j)/2 \rfloor$ Les feuilles de l'arbre correspondent aux intervalles $[x_i, x_{i+1}]$.

Chaque segment est stocké dans le nœud de l'arbre correspondant au plus petit intervalle le contenant.

Utiliser la structure décrite ci dessus pour calculer l'enveloppe inférieure des n segments en temps optimal. On calculera d'abord des enveloppes inférieures avec l'algorithme de la question précédente. Ensuite on les regroupera en $O(\log n)$ paquets dont on calculera l'enveloppe inférieure facilement. Pour finir, on fusionnera ensuite ces $O(\log n)$ enveloppes par l'algorithme division-fusion classique.

Détailler l'algorithme et étudier sa complexité.