Option Informatique

TD $n°05$: distance minimale dans un nuage de points

Il s'agit d'un algorithme de type *diviser pour régner*.

Le problème : On se donne une liste ℓ de longueur n contenant des couples de flottants, représentant un nuage de points, et on veut identifier la distance minimale entre deux de ces points, avec la meilleure complexité possible.

Exercice 1. *Algorithme naïf*

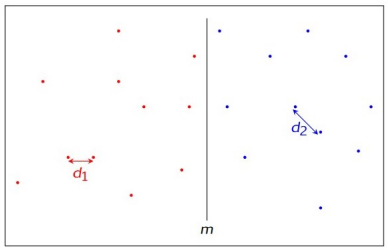
1. Proposer un algorithme quadratique.
2. L'implémenter.

Exercice 2. *Cas de la dimension* 1

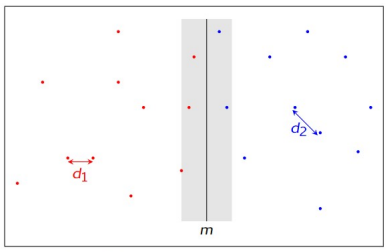
Avant de chercher un algorithme plus performant, étudions un cas particulier très dégénéré : celui où tous les points du nuage se trouvent sur une même droite. Ainsi la liste ℓ représentant le nuage de points sera plutôt ici liste de flottants.

3. Proposer un algorithme en $\Theta(n\lg(n))$. Inutile de l'implémenter.

Exercice 3. *Une stratégie diviser pour régner trop naïve... mais à peine trop.*



Diviser pour régner : on sépare le nuage de points $\mathcal{P}$ en deux parties triées par abscisses croissantes $\mathcal{P}_1$ et $\mathcal{P}_2$ et on calcule par appel récursif les distances minimales d_1 et d_2.



On calcule $\delta = \min(d_1, d_2)$ et on restreint la recherche à la bande verticale délimitée par les abscisses $m-\delta$ et $m+\delta$.

Pour mettre en œuvre cette stratégie efficacement, on réalise un pré-traitement, consistant à trier la liste des points du nuage par abscisses croissantes (plus précisément par ordre lexicographique (abscisse, ordonnée) pour avoir vraiment une relation d'ordre). On obtient une liste ℓx.

Les deux appels récursifs sont alors faciles à réaliser, on utilise les deux sous-listes de ℓx en coupant à la moitié.

De même, il est facile d'obtenir la sous-liste de ℓ constituée des points du nuage situés dans la bande.

Enfin, on peut calculer de façon naïve la distance minimale pour les points de la bande. Dans le meilleur (et le plus probable) des cas, il n'y a que quelques points dans la bande, et ce calcul se fait en $O(1)$. Par contre, dans le pire des cas (même s'il reste exceptionnel), ce calcul est quadratique.

4. Déterminer la complexité de cet algorithme dans le meilleur et dans le pire des cas.

   On ne demande pas de l'implémenter.

Exercice 4. *Stratégie optimale.*

La bonne idée est de réaliser **deux** pré-traitements :

- On trie la liste des points du nuage par abscisses croissantes (plus précisément par ordre lexicographique (abscisse, ordonnée) pour avoir vraiment une relation d'ordre). On obtient une liste ℓx.
- On trie **aussi** la liste des points du nuage par ordonnées croissantes (plus précisément par ordre lexicographique (ordonnée,abscisse) pour avoir vraiment une relation d'ordre). On obtient une liste ℓy.

Les deux appels récursifs sont toujours aussi faciles à réaliser, sur des sous-listes de ℓx.

De même, il est facile d'obtenir la sous-liste de ℓy constituée des points du nuage situés dans la bande.

La sous-liste de ℓy est ainsi triée par ordonnée croissante et on va voir que ceci permet un calcul linéaire de la plus petite distance entre deux points de la bande.

5. Montrer que dans chaque tranche de hauteur δ de la bande ne peuvent se trouver qu'au plus huit points.
   NB : on peut montrer qu'il y en a au plus six mais c'est plus dur.
6. En déduire un algorithme linéaire de calcul la plus petite distance entre deux points de la bande.
7. En déduire la complexité (dans le pire des cas) de l'algorithme obtenu ici.
8. L'implémenter.