7.9 Les deux points les plus rapprochés

Le problème est le suivant :

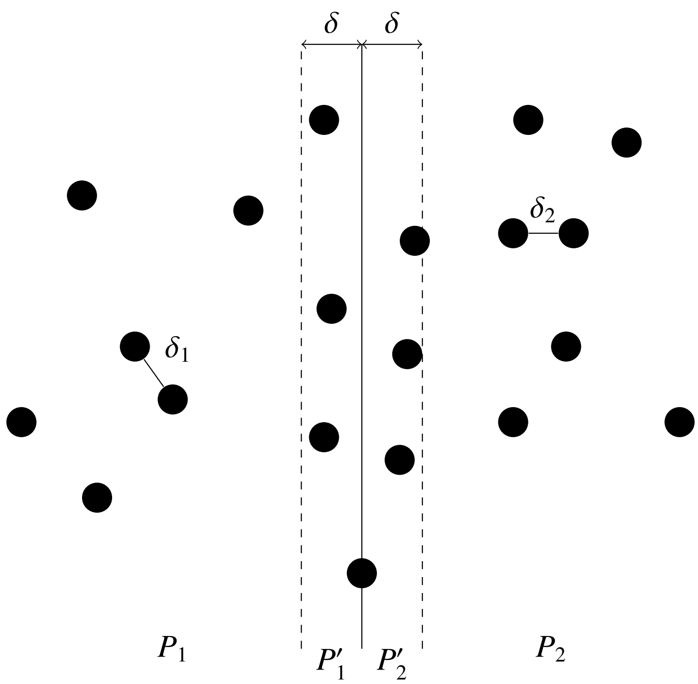
— Entrée : un tableau de points T dans le plan ;

— Sortie : $min_{a,b \in T | a \neq b} d(a, b)$ où $d(a, b)$ est la distance euclidienne entre les points a et b.

L'algorithme naïf est en $O(n^2)$ où n est le nombre de points dans T. Nous proposons ici un algorithme qui repose sur le paradigme 'diviser pour régner'.

L'idée est la suivante : on considère une droite verticale (Δ) qui sépare notre nuage de points en deux parties P_1 et P_2 qui contiennent plus ou moins le même nombre de points. On calcule récursivement la distance minimale δ_1 entre deux points distincts de P_1 et la distance minimale δ_2 entre deux points distincts de P_2. Maintenant, il se peut que la distance minimale entre deux points distincts de T soit atteinte pour un point de P_1 et un point de P_2.

Soit $\delta = min(\delta_1, \delta_2)$. On est amené à calculer la distance minimale entre les points qui sont situés dans une bande verticale centrée en (Δ) et de largeur 2δ.



Pour se faire, il ne faut pas utiliser une approche naïve. L'astuce consiste à représenter les points de cette bande par un tableau P' qui contient les points de cette bande triés par ordonnées croissantes en se rendant compte que pour un point a, il suffit de calculer les distances entre a et b où b est parmis les 7 points qui succèdent a dans P'.

Soit $b_1, \ldots, b_k$ une énumération des points de $P'_1 \cup P'_2$ par ordonnée croissante. On montre qu'il suffit de considérer que les 7 points suivants. Plus précisément :

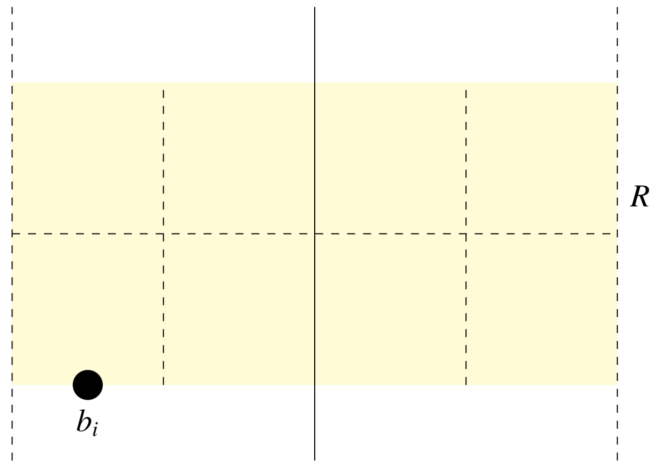
Lemma 3. *Il existe deux points distincts b_i, b_j de $P'_1 \cup P'_2$ avec $d(b_i, b_i) < \delta$*
ssi
il existe i, j, avec $i < j \leq i + 8$ tel que $d(b_i, b_i) < \delta$.

Démonstration.

$\boxed{\Uparrow}$ ok.

$\boxed{\Downarrow}$ Supposons qu'il existe deux points distincts b_i, b_j de $P'_1 \cup P'_2$ avec $i < j$ et $d(b_i, b_i) < \delta$.

Supposons par l'absurde que pour tout i, j, avec $i < j \leq i + 8$ on a $d(b_i, b_i) \geq \delta$. Soit R le rectangle fermé jaune de taille $2\delta \times \delta$ ci-dessous (l'ordonnée de b_i est l'ordonnée du côté bas du rectangle).



b_j est dans R (sinon $d(b_i, b_i) \geq \delta$). Par hypothèse, on $j > i+7$ donc les points $b_i, b_{i+1}, \ldots, b_{i+8}$ sont bien définis. Ils sont dans R et distants de plus δ.

On montre qu'il ne peut pas y avoir plus de 9 points dans R distants les uns des autres d'au moins δ. Pour cela, on découpe ce rectangle en 8 petits carrés de taille $\frac{\delta}{2} \times \frac{\delta}{2}$. Montrons qu'il ne peut y avoir plus d'un point dans un tel carré. Par l'absurde, supposons qu'un tel carré contient deux points. Comme la distance la plus longue du carré est la diagonale de longueur $\sqrt{2}\frac{\delta}{2} < \delta$, ces deux points seront distants de moins de δ. Contradiction. Comme chacun de ces carrés contient au plus un point, le rectangle jaune contient au plus 8 points distants de plus δ. ■

A présent, réfléchissons à l'implémentation. On a deux besoins a priori contradictoire :

- le premier est d'avoir le tableau de points trié par abscisses croissantes afin de pouvoir diviser ce tableau en 2 parties (les points les plus gauches et les points les plus à droite).
- le deuxième est de pouvoir parcourir les points par ordonnées croissantes dans la bande de largeur 2δ.

Trier le tableau à chaque étape selon nos besoins est coûteux ! La solution est donc de stocker nos points deux fois, dans un tableau X qui contient les points triés par abscisses croissantes et dans un tableau Y qui contient ces même points triés par ordonnées croissantes.

Voici l'algorithme présenté entièrement.

Algorithme 26 : plusRapproches(P)

Entrées : Un tableau de points P du plan

Sorties : La distance minimale, i.e. $min_{a,b \in T | a \neq b} d(a,b)$

X := tri(P, par abscisses croissantes) ;
Y := tri(P, par ordonnées croissantes) ;
plusRapprochesRec(X, Y)

Algorithme 27 : $plusRapprochesRec(X, Y)$

Entrées : Deux tableaux X, Y qui contiennent les même points, X est trié par absisse croissante et Y est trié par ordonnée croissante

Sorties : La distance minimale, i.e. $min_{a,b \in X | a \neq b} d(a,b)$

si card(X) ≤ 3 **alors**
    retourner *méthode naïve*
sinon
    $X_G := X[1, \lfloor \frac{|X|}{2} \rfloor]$
    $X_D := X[\lfloor \frac{|X|}{2} \rfloor + 1, |X|]$
    $x_{sep} := X_G[\frac{|X|}{2}].x$
    Y_G := extraire les éléments de Y d'abcsisse $\leq x_{sep}$
    Y_D := extraire les éléments de Y d'abcsisse $> x_{sep}$
    $\delta_G := plusRapprochesRec(X_G, Y_G)$
    $\delta_D := plusRapprochesRec(X_D, Y_D)$
    $\delta := min(\delta_G, \delta_D)$
    Y' := extraire les éléments de Y qui sont dans la bande verticale d'absisse x_{sep} et de largeur 2δ ;
    retourner $min(\delta, plusRapprochesBande(Y', \delta))$

Les fonctions non données ici sont laissés en exercice.