Géométrie algorithmique, examen M1, 7 mars 2008, énoncé

1 Voronoï L_∞

On considère le diagramme de Voronoï de l'ensemble de 6 points sur la feuille jointe pour la distance

$$d(p,q) = max(|x_p - x_q|, |y_p - y_q|).$$

Dessiner la triangulation duale et les « cercles » circonscrits associés (avec des couleurs différentes, ou sur des dessins différents) (et éventuellement le diagramme de Voronoï mais c'est plus difficile).

La feuille jointe à rendre comporte l'ensemble de 6 points dessinés 4 fois (en cas de rature, et pour séparer les différentes réponses si vous voulez).

2 Ordre de points en projection

On considère un ensemble de n points $\mathcal{S}$ du plan. Pour un point $p \in \mathcal{S}$ et un angle $\sigma \in \mathbb{R}$, on note p_σ la projection orthogonale de p sur la droite $D_\sigma : cos(\sigma)x + sin(\sigma)y = 0$.

On oriente cette droite : pour deux points $q_1 = \alpha(-sin(\sigma), cos(\sigma))$ et $q_2 = \beta(-sin(\sigma), cos(\sigma))$ de D_σ, on dira que q_1 est avant q_2 sur D_σ si $\alpha < \beta$.

2.1

Écrire les coordonnées de p_σ en fonction de σ et des coordonnées de p.

2.2

Écrire une condition sur les coordonnées de deux points p et q et sur σ pour que p_σ soit avant q_σ sur D_σ.

Interpréter cette condition géométriquement. Si p et q sont fixés et que σ varie, quand est-ce que l'ordre de p_σ et q_σ change ?

2.3

Si on fait varier σ et que l'on a n points, combien d'ordres sont possibles pour les n points (au maximum). Ou, pour être plus précis, en combien de valeurs σ l'ordre change-t-il ?

2.4

Proposer un algorithme pour construire une structure de données représentant l'ordre des points de $\mathcal{S}_\sigma = \{p_\sigma; p \in \mathcal{S}\}$ pour toutes les valeurs de $\sigma \in [0, \pi]$ (écrire du pseudo-code).

Ensuite on effectue des requêtes : on donne une valeur de σ et on doit récupérer l'ordre de $\mathcal{S}_\sigma$. Expliquer comment une telle requête est traitée par votre structure de données (écrire du pseudo-code).

2.5

Donner la complexité du calcul de la structure et la complexité d'une requête.

[Remarque pour 2.4 et 2.5 : donner l'ordre des points de $\mathcal{S}_\sigma$ prend au moins un temps n (il faut énumérer les points), on essaiera donc de se rapprocher de $O(n)$ pour la complexité d'une requête.]

3 Graphe des carrés vides

On considère un ensemble de n points $\mathcal{S}$ du plan.

On construit un graphe G_1 en reliant deux points p et q de $\mathcal{S}$ si le carré $[x_m - \frac{d}{\sqrt{2}}, x_m + \frac{d}{\sqrt{2}}] \times [y_m - \frac{d}{\sqrt{2}}, y_m + \frac{d}{\sqrt{2}}]$ ne contient pas de points de $\mathcal{S}$, où $d = \frac{|pq|}{2}$, m est le milieu de pq et $|pq|$ est la notation pour la distance euclidienne entre p et q.

On construit également un graphe G_2 en reliant deux points p et q de $\mathcal{S}$ si le carré $[x_m - d, x_m + d] \times [y_m - d, y_m + d]$ ne contient pas d'autres points de $\mathcal{S}$ que p et q.

3.1

Montrer que les arêtes de G_2 sont des arêtes de Delaunay.

Montrer que les arêtes de Gabriel sont des arêtes de G_1.

3.2

Donner une borne inférieure et supérieure sur la taille de G_1, et des exemples réalisant ces bornes pour un ensemble de n points $\forall n$.

[On se contentera d'ordres de grandeurs : $\Omega(?)$, $O(?)$]

3.3

Donner une borne supérieure sur la taille de G_2, et des exemples réalisant cette borne pour un ensemble de n points $\forall n$.

Montrer que G_2 n'est pas vide (trouver une arête).

4 Rectangles

On se donne $\mathcal{R}$ un ensemble de n rectangles $r = [x_r, X_r] \times [y_r, Y_r]$. On suppose que tous les rectangles contiennent l'origine : $x_r, y_r < 0 < X_r, Y_r$.

4.1

Proposer un algorithme pour calculer l'intersection des rectangles de R et donner sa complexité.

4.2

Montrer que le bord de l'union des rectangles (quand on parcours le bord dans le sens direct) de R peut être découpé en 4 parties avec :

— x décroissant et y décroissant,

— x décroissant et y croissant,

— x croissant et y décroissant et

— x croissant et y croissant.

4.3

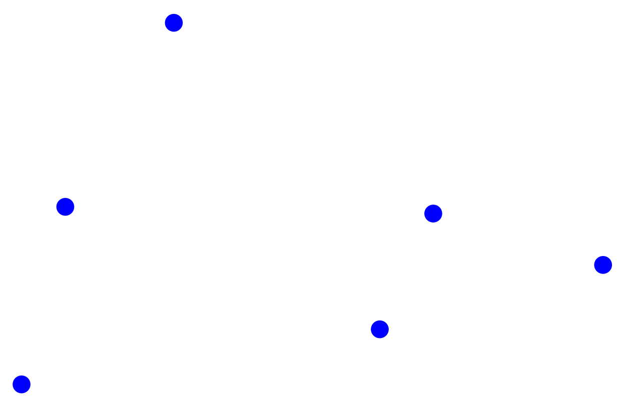
On suppose maintenant que les coordonnées des rectangles sont des nombres aléatoires indépendants dans $[0, 1]$ pour X_r et Y_r et dans $[-1, 0]$ pour x_r et y_r.

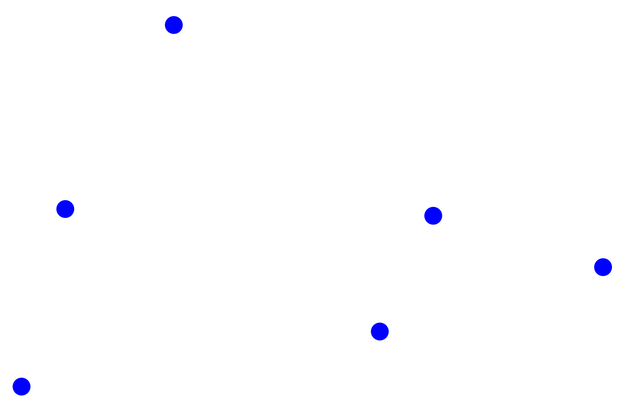
Quelle est la probabilité qu'un rectangle r de $\mathcal{R}$ apparaisse sur le bord de l'union dans la partie « x décroissant et y croissant » ? On pourra considérer une partition de $\mathcal{R}$ entre les rectangles t avec $X_t < X_r$ et ceux avec $X_t > X_r$. Si le nombre de rectangles t du deuxième paquet est $i - 1$, on pourra exprimer cette probabilité en fonction de i.

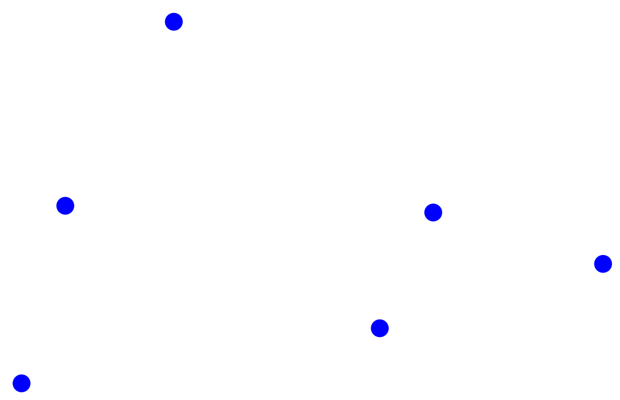
En déduire la taille moyenne de cette union de rectangles.

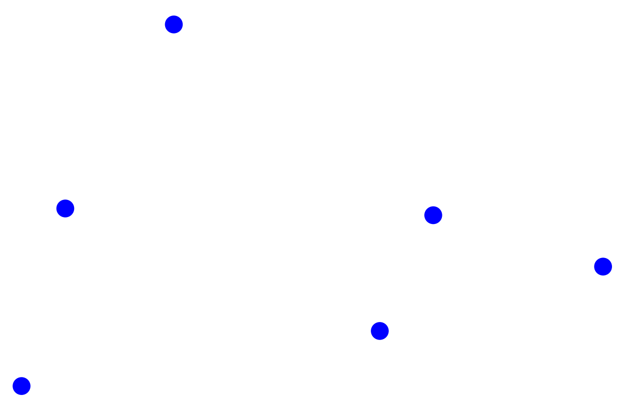
Examen Master STIC, géométrie algorithmique 2007-2008
Figure à compléter et à rendre

Nom :
Prénom :

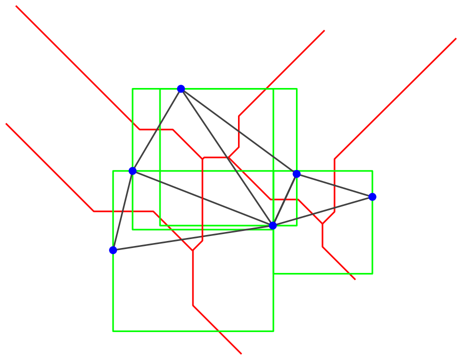








1 Voronoï L_∞



2 Ordre de points en projection

2.1

$p_\sigma = p + \lambda(cos(\sigma), sin(\sigma))$ pour un certain λ et doit vérifier l'équation de D_σ.
d'où $\lambda = -(cos(\sigma)x_p + sin(\sigma)y_p)$.

2.2

Il faut que la projection du vecteur $\vec{pq}$ sur D_σ soit bien dans le sens positif de D_σ.
Soit $\vec{pq} \cdot (-sin(\sigma), cos(\sigma)) > 0$.

L'ordre change si la direction du vecteur $\vec{pq}$ est orthogonale à D_σ. On note σ_{pq} l'angle $\widehat{Ox, \vec{pq}}$

2.3

Chaque paire de points définit un angle σ_{pq} il y a $\frac{n(n-1)}{2}$ telles paires de points.

2.4

La structure consiste en l'ensemble trié de tous les angles critiques, l'ordre associé est stocké pour certains angles critiques (ceux qui ont un rang multiple de n).

On calcule et on trie les points de $\mathcal{S}_0$ et on stocke l'ordre dans un tableau T_0 ;
On calcule et on trie les valeurs σ_{pq} que l'on stocke dans un tableau A ;
Chaque angle σ_{pq} a un pointeur vers p et q
On recopie T_0 dans une liste doublement chaînée L ;
chaque point p a un pointeur vers sa position dans L
i=0 ;
Pour $i = 0$ à $\frac{n(n-1)}{2}$
    Soit p et q tel que $A[i] = \sigma_{pq}$
    échanger p et q dans L *(ainsi L représente toujours l'ordre courant)*
    si i multiple de n recopier l'ordre L dans T_i.
    i = i+1 ;

Maintenant le traitement d'une requête σ
Trouver i tel que $A[i] \le \sigma < A[i+1]$; *(recherche dichotomique)*
Soit j le multiple de n immédiatement inférieur à i ;
Recopier l'ordre T_j dans une liste L ;
Pour $k = j$ à i
    Soit p et q tel que $A[k] = \sigma_{pq}$
    échanger p et q dans L *(ainsi L représente toujours l'ordre courant)*
    k = k+1 ;
Renvoyer L ;

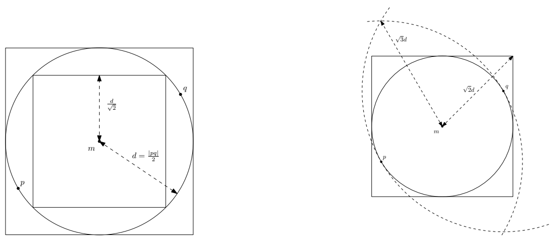
2.5

Prétraitement : $O(n^2 \log n)$.
C'est le tri de A qui est le plus limitant. La boucle a clairement une longueur quadratique et chaque passage prends un temps constant.

Requête : $O(n)$.
La boucle a une taille inférieure à n qui était l'objectif visé. Un résultat supérieur ou égal à $O(n \log n)$ n'a pas grand sens puisque c'est ce qu'on obtient par projection et tri direct.

3 Graphes des carrés vides

3.1



Le cercle de diamètre pq est encadré par les deux carrés construits (Fig gauche).

3.2

Gabriel est inclus dans G_1 donc la taille minimale est de $\Omega(n)$ arêtes. Gabriel a une taille $\Omega(n)$ car le nombre d'arêtes est $3n - k - 3$ et celui de triangles est $2n - k - 2$; il y a au plus un angle obtus par triangle et il y a un angle obtu par arête de Delaunay pas de Gabriel. Exemple : des points sur l'axe des x plus un point en dehors.

La taille maximale est $O(n^2)$. Exemple : $p_i = (0, i\varepsilon)$ pour $i < \frac{n}{2}$ et $p_i = (2, 1 - i\varepsilon)$ pour $i \geq \frac{n}{2}$ avec ϵ assez petit ($< \frac{1}{2n}$). Alors n'importe quel point du premier paquet est relié à n'importe quel point du deuxième paquet.

3.3

G_2 est inclus dans Delaunay donc la taille maximale est $O(n)$. Exemple : des points régulièrement espacé sur l'axe des x plus un point loin en dehors, toutes les arêtes sur l'axe des x sont dans G_2.

La plus proche paire de $\mathcal{S}$ est une arête de G_2. Le carré définissant G_2 est inclus dans le cercle de centre m et de rayon $\sqrt{2}d$, le cercle de centre m et de rayon $\sqrt{3}d$ est inclus dans l'union des cercles de centres p et q et de rayon $2d$ vides par hypothèse que pq est la plus proche paire (Fig. droite).

4 Rectangles

4.1

L'intersection de 2 rectangles est un rectangle et se calcule en temps constant :

$$r \cap t = [max(x_r, x_t), min(X_r, X_t)] \times [max(y_r, y_t), min(Y_r, Y_t)]$$

l'intersection n'est pas vide car elle contient l'origine.

Un algorithme incrémental a donc trivialement une complexité $\Theta(n)$.

4.2

Si on découpe r en $[0, X_r] \times [0, Y_r] \cup [x_r, 0] \times [0, Y_r] \cup [x_r, 0] \times [y_r, 0] \cup [0, X_r] \times [y_r, 0]$ on a 4 problèmes indépendants dans les 4 quadrants.

Dans le quadrant $x, y > 0$ le bord de l'union a un segment sur chaque axe de coordonnée et une portion en escalier à x décroissant et y croissant. En effet quand on tourne autour d'un rectangle dans le sens direct le bord droit est parcouru à y croissant et le bord gauche à y décroissant, mais dans ce quadrant on a que des bords droits et donc que du y croissant (et similairement que des bords haut et donc que du x décroissant).

4.3

On regarde donc d'abord la contribution dans le premier quadrant.

Si r correspond au i-ème bord droit en partant de la droite, r contribuera au bord de l'union si les $i - 1$ triangles plus à droite ont tous des coordonnées Y_t inférieure à Y_r. En d'autre termes, il faut que Y_r soit le plus grand de i valeurs indépendantes ce qui se produit avec une probabilité $\frac{1}{i}$. Quand on réunit avec les autres quadrants on multiplie par 4 et on obtiens $\frac{4}{i}$.

La contribution de tous les rectangles est alors la somme sur i de ces valeurs et on a $O(\log n)$ pour la taille moyenne de l'union des rectangles.