

# TP2 - Petit arboretum.

## Exercice 3 - Quelle est la ville la plus proche de Ponts ?

Yonatan DELORO

Pour le 7 décembre 2016

Dans ce TP, on cherche à stocker efficacement les coordonnées des villes françaises, pour faciliter la recherche de plus proches voisins. On utilise pour ce faire la structure du quadtree.

### 1 Q2. Construction du quadtree

On crée une feuille vide puis on insère successivement chaque point. Le principe général de l'algorithme d'insertion d'un point est donné dans les transparents du cours : on "réduit" récursivement les coordonnées du point à insérer (pour ce faire utilisation des "quadrants") jusqu'à atteindre une feuille. Arrivé à une feuille : soit la case à vide et on y insère le point, soit elle est occupée et on la redécoupe récursivement jusqu'à ce que les deux points occupent deux cases distinctes. Les commentaires présents dans le code détaillent les étapes de l'algorithme.

### 2 Q3.Q4. Recherche des voisins à une distance $r$ d'un point, et de ses plus proches voisins

On implémente une version récursive de la recherche des voisins à une distance  $r$  d'un point  $p$ . On part de la racine de l'arbre et on examine chacun de ses fils. Si le fils est une feuille, on peut directement calculer la distance du point contenue dans la case à  $p$ . Sinon, afin que le sous-arbre que constitue le fils contienne des voisins tels que recherchés, il est nécessaire que sa case intersecte la boule de rayon  $r$  centrée en  $p$ . C'est l'intérêt du stockage des points dans le quadtree : limiter le nombre de points visités au cours de la recherche en regroupant au mieux les points. Un quadtree est ainsi d'autant plus efficace que les points sont densément répartis sur la zone géographique donnée.

Pour la recherche du plus proche voisin, il suffit de rechercher les voisins à une distance inférieure à  $r$  du point  $p$  (prise au départ à l'infini), puis à actualiser  $r$  dès qu'un plus proche voisin de  $p$  est rencontré.

Pour tester si une boule centrée en  $p$  de rayon  $r$  intersecte notre case carrée, on calcule les coordonnées, horizontale et verticale, du projeté du centre du cercle sur le carré, qui est le point le plus proche du cercle appartenant au carré. Ainsi il ne reste plus qu'à comparer à  $r$  la distance de ce projeté au centre du cercle.

Les commentaires du code expliquent en détail l'algorithme de recherche de plus proche voisin.

### 3 Q6. Quadtree des villes de France métropolitaine

Le quadtree des villes est composé de 34095 feuilles et de 20961 feuilles (taille de 55056). Sa profondeur est de 15.

### 4 Q7. Ville la plus proche de "Ponts"

La ville la plus proche de Ponts est Saint-Jean-de-la Haize, situé à 1.227 km de Ponts. 162 points sont visités au cours de la recherche dans le quadtree. Avec un parcours linéaire du vecteur des villes, on aurait dû parcourir la totalité des villes, soit 35181 points. On visite donc avec un quadtree moins d'un demi pour cent du nombre de villes visitées avec un parcours linéaire.

### 5 Q8. Temps moyen pour trouver une plus proche ville

On sélectionne pour ce faire un échantillon de 100 villes tirées au hasard, et on recherche leurs plus proches voisins. Avec un quadtree, on obtient, sur cet échantillon, un temps moyen de recherche de plus proche voisin de l'ordre de  $6,4 \cdot 10^{-4}$  secondes. Avec un vector (parcours linéaire), on obtient un temps moyen de  $2,3 \cdot 10^{-2}$  secondes.

### 6 Q9. Rentabiliser la construction d'un quadtree

Le temps de construction du quadtree est de 0,214 secondes. Ainsi le nombre  $N$  de recherches de plus proches villes qu'il faut faire en moyenne pour rentabiliser le temps de construction du quadtree est donné par :

$$\text{temps\_construction\_quadtree} + N * \text{temps\_moyen\_recherche\_quadtree} = N * \text{temps\_moyen\_recherche\_vector}$$

Soit :

$$N = \frac{\text{temps\_construction\_quadtree}}{\text{temps\_moyen\_recherche\_vector} - \text{temps\_moyen\_recherche\_quadtree}}$$

On obtient sur notre échantillon  $N = 9,78$ . On rentabilise la construction du quadtree après 10 recherches en moyenne de plus proches voisins.