Rapport : Algorithmique (projet)

Introduction :

Le projet consiste à créer de diverses composantes connexes, pour un ensemble de points situés dans le plan [0,1]. La relation ici déterminée ne serait qu'une distance inférieure à la distance seuil, qui permettrait de relier un point à un autre.

Moments forts :

1) Réflexion et choix de l'algorithme :

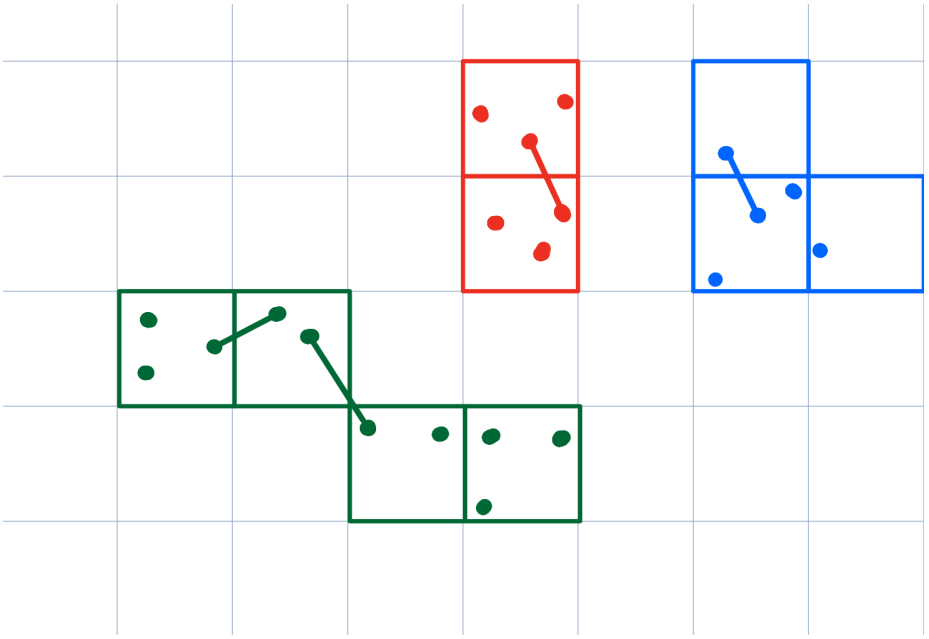
Première approche et points faibles :

La simple lecture du sujet suggère l'idée du parcours des graphes. Une première approche (rejetée) était de parcourir chaque point et de l'associer à un autre par le biais de la relation distance donnée.
Une telle approche donnait certes le résultat désiré, mais en un temps trop long, vu la nécessité d'un parcours double de l'ensemble de tous les points (complexité dominée par le carré du nombre total de points).

Problématique et méthodologie:

Nous avons ainsi décidé de tout modifier. En effet, la question était : au lieu de gérer un immense nombre de points dans le plan, n'y a-t-il pas d'autre moyen de réduire le nombre de répétitions des opérations ?
L'idée était alors de créer des blocs contenant des points, et puis ne travailler que sur ces blocs-ci. Le choix des blocs dépend forcément de la distance seuil. La meilleure subdivision du plan pour nous était une forme de quadrillage du plan, en divisant ce dernier en des carrés dont leur diagonale n'était autre que la distance, puisque la diagonale d'un carré est par définition la plus grande longueur. Ainsi, travailler sur un carré revient à travailler sur un ensemble de points qui sont **forcément**

liés, c'est-à-dire que la distance entre eux est nécessairement inférieure à la distance seuil, en initialisant chaque carré dans une matrice sous forme de sets contenant des points. Le choix de la forme SET sera justifié dans la gestion de mémoire traitée ci-dessous .



Ensuite, un point dans un carré particulier peut être lié à un point dans un autre carré, puisque l'arête de ce carré est la distance seuil divisée par la racine de deux. Le schéma en haut indique qu'il faut alors effectuer une recherche dans les carrés adjacents, qui sont les deux carrés situés à gauche, à droite, en haut et en bas de celui-ci. Enfin, on effectue le parcours dans le graphe dont les éléments sont bien les carrés, et la relation étant celle définie juste avant.

D'ailleurs, aucune condition sur les entrées n'a été ajoutée pour l'exécution du programme.

2) Gestion de mémoire :

Volet hachage :

Nous avons mentionné en haut que les carrés sont tous assemblés dans une grande matrice. Comme la recherche dans ces carrés est primordiale après, il faut alors choisir une forme qui permet d'effectuer la recherche avec un coût minimal. Ceci donne alors l'idée d'utiliser des SETS

dans python pour minimiser le coût des opérations effectuées lors du parcours des carreaux, au lieu de l'utilisation des listes. Un carré est ainsi un set qui contient des points forcément reliés, tous les sets étant stockés dans la matrice de départ.

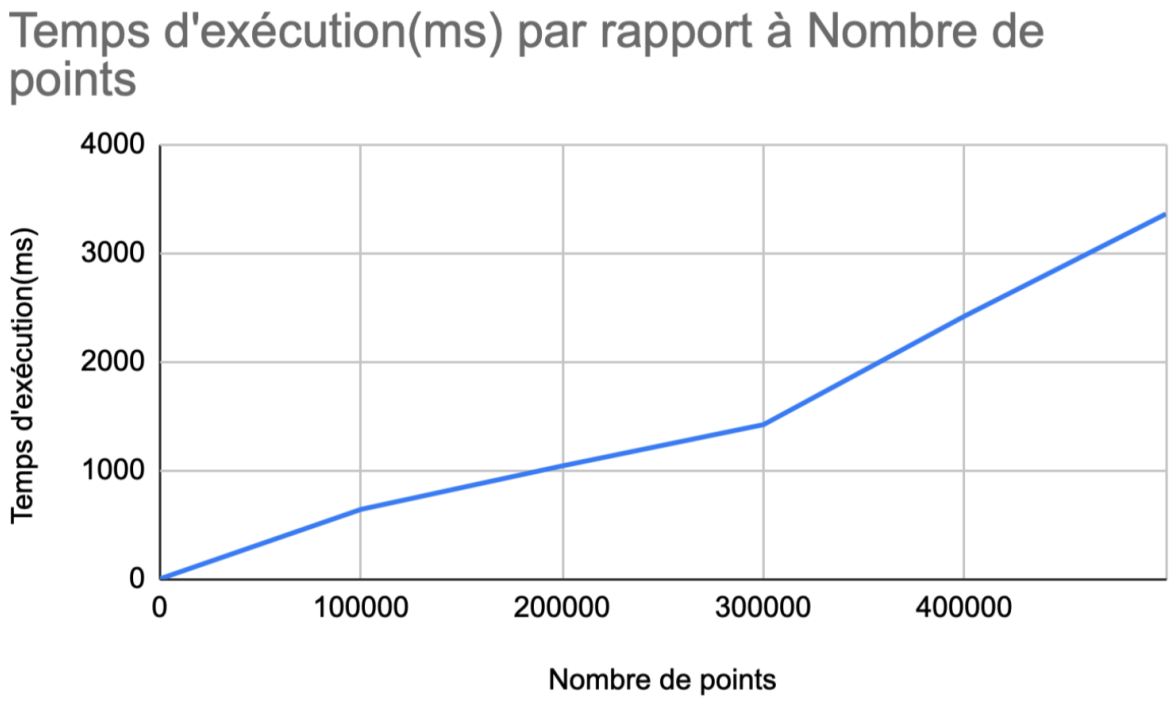
Volet complexité :

Quant à la gestion de la mémoire, la complexité de l'algorithme implémenté dépend initialement du nombre de carrés dans l'espace, qui est

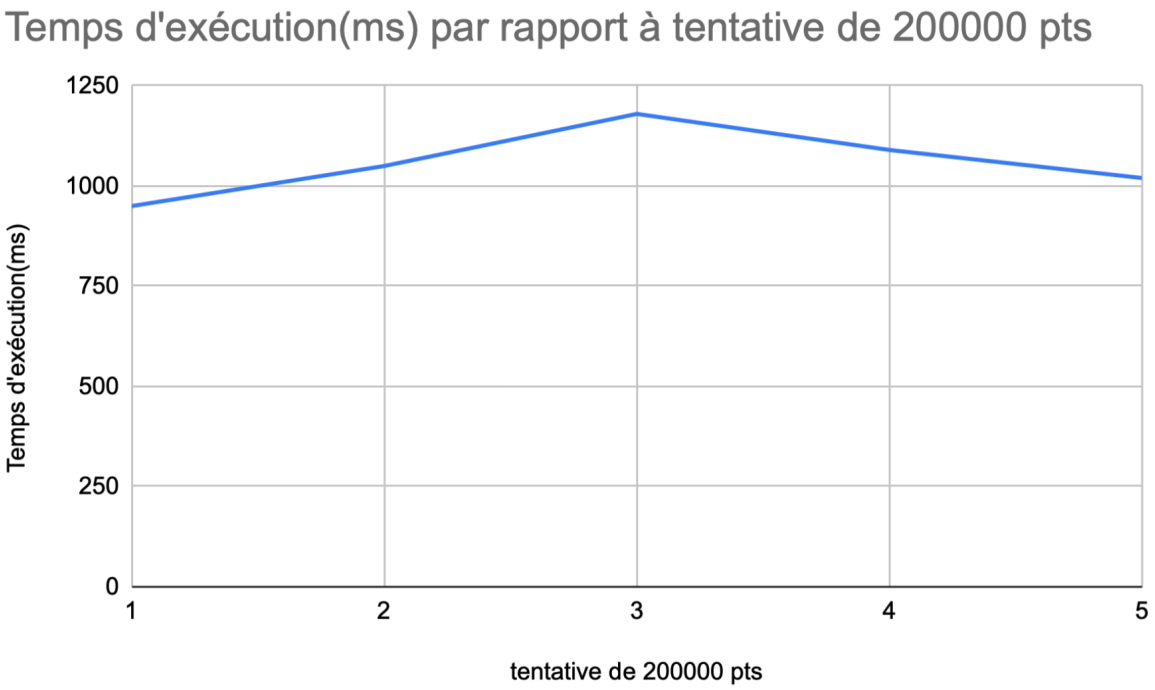
$$E(\frac{\sqrt{2}}{distance})$$

où E désigne la partie entière, en addition avec le nombre des points qui sont en relation dans un carré avec ceux d'un autre. Cette somme n'est proche du nombre total de points que dans des cas rares, où ces points sont condensés dans un carré particulier. Le traitement dans ce cas est aussi de complexité basse.

Avec des tests de points générés aléatoirement, on obtient la courbe suivante en testant sur 100 000 jusqu'à 500 000 points.



Pour vérifier la légitimité de mes tests, voici la variation du temps d'exécution de notre programme en générant 200 000 points à reprise.



Conclusion :

Ce projet nous a permis de développer mes compétences en algorithmique, en programmation et aussi au niveau de l'imagination en point de vue réflexion. Nous avons appris à optimiser des algorithmes, pour avoir un temps d'exécution raisonnable avec une bonne complexité. Le but formatif a été amplement atteint. Tous nos remerciements aux responsables du projet et du module.