Exercice de programmation

(CSI 2510)

Expérience 1 :

Après avoir exécuté la classe « Exp1.java » pour chacun des points donnés, 2 fichiers textes ont été obtenus par point. L'un des fichiers contenait les points voisins obtenus avec la méthode linéaire, tandis que l'autre contenait tous les points voisins obtenus avec la méthode de l'arbre K-D. Les deux fichiers contenaient des résultats similaires puisqu'ils avaient le même nombre de points et, lorsqu'on les compare avec VSCode, ils contenaient les mêmes points mais dans un ordre différent.

Expérience 2 :

La classe « Exp2.java » a été exécuté deux fois pour cette expérience et, à chaque fois, on a utilisé un bond de 10 entre chaque point. D'abord, la méthode linéaire a été testé et nous avons obtenus un temps de calcul moyen de 0.333726 ms (Point_Cloud_1.csv). Ensuite, la méthode de l'arbre K-D a été testé nous avons obtenus un temps de calcul moyen de 0.071683 ms (Point_Cloud_1.csv). Nous pouvons donc remarquer que la dernière méthode est plus efficace, ce qui était attendu puisque sa complexité est O(log(n)) alors que la complexité de la méthode linéaire est O(n).

Expérience 3 :

TEMPS D'EXÉCUTION (ms)	**DBScan (Linéaire)**	**DBScan (Arbre K-D)**
Point_Cloud_1.csv	9 269.745	6 119.28
Point_Cloud_2.csv	22 157.1649	13 748.6501
Point_Cloud_3.csv	14 136.4891	11 249.9374

Comme on peut le remarquer, la méthode avec l'arbre K-D est plus rapide que la méthode linéaire puisqu'elle est plus efficace. De plus, les fichiers obtenus à l'aide des deux méthodes contiennent le même nombre de « clusters » et « noise points » ce qui était attendu.