SAE 2.02 – Exploration algorithmique d’un problème

Par Ludmann Dorian et Maupou Cassandra 1.3b

À la suite de simples tests, la limite de nos fonctions a été remarquable par le fait qu’elles ne pouvaient traiter qu’un nombre limité de données avant que le temps d’exécution considère nos fonctions comme inutilisables.

Pour réussir à implémenter les requêtes à grande échelle, il faut alors factoriser le code. Pour cela, plusieurs façons s’offrent à nous :

* Utiliser des attributs dans le dictionnaire du graphe avec : prédécesseur, distance avec prochain point, déjà visité…
* Utiliser un système de sauvegarde pour éviter de recalculer des valeurs déjà connus (Exemple de la question 7)
* Utiliser des fonctions de la bibliothèque networkx (algorithme de Dijkstra)

L’ensemble de ces solutions peuvent exercer une grande influence quant au temps d’exécution des fonctions, mais ne reste pas suffisante comparé à la masse de données du problème du nombre de Bacon.

La dernière solution serait alors de changer la modélisation du graphe (en utilisant une autre librairie par exemple) ou alors changer de langage de programmation.

Pour pouvoir améliorer le temps d’exécution de nos fonctions, il nous faudrait alors plus de temps mais surtout plus de connaissances. Malgré le nombre d’informations disponibles sur Internet, de grands problèmes mathématiques restent très peu développés voir non résolus, comme trouver le chemin non cyclique le plus long entre deux sommets, que le graphe soit pondéré ou non.