

## SAE Algo - À la conquête d'Hollywood

### Rapport individuel

Pour cette SAE, j'ai pu commencer à répondre aux questions et à développer les méthodes en Java. Je me suis principalement occupé du début, avec les trois premières méthodes : `collaborateursEnCommun`, `collaborateursProches` et `distanceActeur`. Le reste a été fait par mon camarade.

Au début, j'ai eu du mal à me lancer, car en utilisant la bibliothèque `JGraphT`, j'ai perdu du temps inutilement : j'avais commencé à réécrire des méthodes moi-même alors que la librairie les proposait déjà... Cela m'a fait perdre un peu de temps.

Concernant la première méthode, rien n'était très compliqué : il suffisait de récupérer les voisins de `v1`, puis ceux de `v2`, et de les comparer. Cette méthode s'est donc bien passée.

Pour la deuxième méthode, qui permet de trouver tous les sommets (collaborateurs) à une distance inférieure ou égale à  $k$  d'un sommet dans le graphe, j'ai utilisé un parcours en largeur. C'est un type de parcours qu'on avait très peu vu en cours, donc j'ai dû faire des recherches pour savoir comment l'utiliser avec `JGraphT`. Heureusement, il y a pas mal de documentation, donc ça allait.

Le seul problème que j'ai eu, c'est que je n'avais pas créé un ensemble pour vérifier si un nœud avait déjà été visité. Du coup, le code marchait, mais il était très, très, très lent.

Pour la troisième méthode, ça a été plus facile car il fallait calculer la distance entre deux acteurs. Je n'ai pas eu de problème avec cette méthode, car je commençais à bien comprendre les méthodes et la bibliothèque `JGraphT`. De plus, nous avons eu beaucoup plus de cours sur ce sujet, ce qui m'a beaucoup aidé. Je n'ai donc pas rencontré de difficultés pour cette méthode.

Démonstration de compétence et acquisition des compétences :

Pour cette SAE, j'ai pu mobiliser beaucoup de connaissance et compétences que j'ai apprises cette année. J'ai utilisé mes compétences en Java, combinées au TP sur les graphes, qui nous a beaucoup aidés pour cette SAE, car c'était un peu comme un TP, mais plus complexe. J'ai aussi utilisé mes connaissances en qualité de développement pour gérer le dépôt GitHub. Nous avons dû refaire un dépôt final, car celui que nous avons créé au fur et à mesure posait trop de problèmes de conflits.

Pour cette SAE, j'ai appris à analyser un problème complexe en le découpant en étapes simples et en choisissant les structures de données adaptées, comme les files et les ensembles pour le parcours en largeur. J'ai comparé différents algorithmes, notamment le parcours en largeur et le parcours en profondeur, et compris que le parcours en largeur est plus efficace pour trouver la distance minimale dans un graphe.