

Pour ce qui est de tous ces algorithmes j'avais commencé à les faire en jquery mais à cause d'un problème d'implémentation de celui-ci à un certain moment j'ai décidé de tout re-coder en js pure.

Pour ce qui est des scripts qui font l'interfaçage avec la base de donnée en ajax je les ai fait tout à la fin et ils ne servent qu'aux deux premiers algorithmes et seront modifiés si l'on veut l'utiliser avec autre chose. Je rajoute que pour moi la partie la plus difficile était celle là car je n'avais jamais utilisé ajax et je souhaite ne plus jamais avoir à l'utiliser. Le premier crée une url et appelle la fonction ajax avec cette url fraîchement créée afin que l'on puisse exécuter le php ce trouvant dans le fichier pourbdd.php.

Les langages utilisés sont donc l'Ajax et le php car on n'avait initialement pas le droit d'utiliser le react et que le php a été un peu forcé. Il me restait donc que l'ajax pour pouvoir produire quelque chose d'un peu dynamique

## algo1:

Pour cet algorithme j'ai dû créer plusieurs fonctions. Tout d'abord, j'ai créé une constante `byld` afin de me faciliter la tâche en js pure. Je me suis après attaqué à créer l'algorithme. J'ai d'abord créé la fonction recherche qui génère des requêtes sql approximatives en fonction de ce qu'a rentré l'utilisateur. Pour ce faire, j'ai dû trouver une fonction pour remplacer les accents et caractères spéciaux indésirables. Une fois que tout cela était fait je me suis occupé de créer une fonction qui fait en sorte que l'on envoie des requêtes à la base de donnée seulement si l'utilisateur a entré au moins 5 caractères dans la barre de recherche afin de ne pas surcharger la base de donnée.

J'ai choisis le Javascript pour faire cela car le javascript est mon langage de prédilection mais finalement je pense déporter une partie en PHP afin de gagner en sécurité

## algo2:

Pour ce qui est du second algorithme, il est encore plus simple que le premier. En effet, il va juste chercher les deux saisies de l'utilisateur et récupère tous les films ou acteurs puis compare ceux qui ont joué dedans ou tous les films dans lesquels ils ont joué et ne garde que les points communs qu'il affiche. S'il n'y a pas de point commun il affiche qu'il n'y en a pas.

Les langages sont les mêmes que l'algo 1 et je migrerais sûrement en php pour les mêmes raisons

## algo3:

Pour le 3eme algo il a fallu créer deux scripts: un script qui crée un graphe et le place dans un fichier. De ce côté j'ai personnellement écrit un script qui ne comprend même pas le graphe mais qui se contente de le créer dans un autre fichier. Pour la seconde partie j'ai créé un script qui utilise l'algorithme BFS qui permet de trouver le plus court chemin entre 2 sommets d'un graphe non ordonné. Pour faire ces algos j'ai choisi le python car c'était un langage plutôt approprié pour faire cela.

En effet, un langage procédural exécutable facilement sur une machine me paraissait parfait pour créer un graphe. J'ai tout d'abord essayé de créer l'autre algos en js sans passer par un graphe mais cela était un enfer et c'est pour cela que mon choix s'est finalement porté sur le python.

Pour ce qui est des bibliothèques utilisées pour la connexion a la base de donnée j'ai utilisé la solution fournis par mon camarade claud et pour la bibliothèque utilisé dans le bfs il se trouve qu'après pas mal de recherche c'est la solution la plus simple que j'ai trouvé.