

## Recherche de chemin

### ATTENTION : NE PAS OUVRIR LE FICHIER DATA.PHP

Le but de cet exercice est de se baser uniquement sur les fonctions visibles dans user.php, sans dépendre ni se fier à leur implémentation dans data.php. Ainsi, si son fonctionnement venait à changer, votre code devrait fonctionner immédiatement, sans toucher une seule ligne, en remplaçant simplement data.php par la nouvelle version.

De même, évitez de chercher l'algorithme sur internet, c'est à vous de l'écrire.

### L'exercice :

L'objectif est d'écrire un algorithme permettant de trouver le chemin le plus court dans un graphe. Pour rappel, un graphe est un ensemble de noeuds(nodes) reliés par des arrêtes(edges) de longueur différentes. Cela peut sembler abstrait, mais un graphe peut représenter beaucoup de choses, d'une carte (les noeuds étant les carrefours, les arêtes les routes) à un réseau de connaissances (les noeuds des personnes, avec des arrêtes pour relier celles qui se connaissent). Dans tout les cas, faire transiter efficacement des informations (ou tout autre choses) d'un noeud à un autre est un problème récurrent.

Pour le résoudre, vous allez, dans le fichier user.php, écrire la fonction FindPath. Vous pourrez utiliser les fonctions déjà présentes pour accéder aux données, mais vous n'avez pas le droit d'utiliser directement le contenu de data.php.

Vous pourrez vérifier vos résultats sur index.php, qui affiche de manière graphique le retour de FindPath.

Vous avez un début de projet comprenant 3 fichiers :

- data.php : Ne l'ouvrez pas
- index.php : Le fichier qui utilisera votre code. Vous pouvez regarder comment il utilise votre fonction, mais il doit fonctionner sans être modifié!
- user.php : Le fichier dans lequel vous allez travailler, et qui expose des fonctions qui vous seront utiles

Voici quelques directives pour écrire l'algorithme :

- Il se base sur une liste de noeuds déjà visités, et une liste de noeuds à visiter(cette dernière sera initialement composée du noeud de départ uniquement)
- Chaque noeud à visiter aura un coût : c'est la distance entre le départ et le noeud en question (plus précisément, la somme des distance entre le départ et tous les noeuds jusqu'à celui-ci)
- On visite toujours le noeud avec le plus faible coût en priorité.
- À chaque noeud visité, tous ses voisins seront à visiter.
- Si on tente d'ajouter un noeud à la liste à visiter, mais qu'il y est déjà présent, on retiendra uniquement la version qui a le plus faible coût.
- On ne revisite pas un noeud déjà visité.
- Afin de retracer le chemin, on doit savoir pour chaque noeud visité et à visiter le noeud parent par lequel on est arrivé.
- Une fois le noeud d'arrivée visité, il suffit de remonter la chaîne des parents pour recomposer le chemin.
- Vous n'êtes pas obligés de suivre ces directives, mais c'est fortement conseillé!