PROJET ALGORITHMIQUE (1)

Gestion d’un réseau de bus

Suivre tuto git yt

15/03/2022

* 1 branche git par tâche
* Politique de nomination, nom des branches (Tache\_1\_nom\_de\_la\_tache)

Séparation des tâches.

Github.

On peut essayer déjà de manipuler la liste d’horaires donnée. La séparer afin de pouvoir tester le code qu’avec une seule liste d’arrêt (exemple prendre les deux premiers arrêts du fichier et leurs horaires.)

* Structure de données :

Node Arrêt\_de\_bus : contient ses horaires ? +ses enfants (selon la ligne de bus) nom d’arrêt ?

Chaque arrêt devra pouvoir indiquer vers quel arrêt on peut aller avec quelle ligne de bus.

* ARRET1 ([Arret0, 10min, ligne1], [Arret2, 15min, ligne1], [Arret3, 5min, ligne2])
* Ne marche pas car liste de plusieurs choses différentes
* ARRET1([liste des horaires], [listes des horaires\_jours\_fériés], [ARRET0, ARRET2,ARRET3], [10min, 15min, 5min], [ligne1, ligne1, ligne2]) On doit faire selon les heures de depart et d arrivee
* ARRET1([liste des horaires], [listes des horaires\_jours\_fériés], [ARRET0, ARRET2,ARRET3], [ligne1, ligne1, ligne2])

Ici, le poids de l’arc sera heure de départ – heure d’arrivée : il faudra prendre en compte l’ordre de la liste pour savoir quelle liste prendre. On pourra déjà essayer avec une seule liste d’horaire.

Un Node se définit avec ses horaires et son nom. (Pas avec la ligne de bus car il peut être su plusieurs lignes)

If node already exist ? Parcours du graphe à chaque itération lors de l’implémentation ? 🡪 add voisin

Graphe réseau SIBRA ?

* Méthodes

Add voisin remove voisin 🡪 ajout de l arrêt voisin avec durée et ligne

Différents alogrithmes de plus court chemin : shortest (juste nombre arrets) (pas besoin du poids des arcs),

Fastest (duree) ,

foremost (arriver le plus tot)

Dijkstra ?(cours graphe)

AFFICHAGE DIRECTION

« de ARRET1 à ARRET2, prendre la ligne n jusqu’à ARRET3 puis prendre la ligne X jusqu’à ARRET2 »

* Implémentation partie de sibra (fin)
* Data avec liste des horaire pour chaque arrêt : type horaire ? On a le code avec la liste des horaires pour chaque arret🡪 boucle for avec liste et nom d horaire, on a un fichier par ligne de bus
* Pour chaque arret on a undictionnaire avec des listes

HEURES EN THEORIE 8 (ca va être short=):

1h : séparer la liste donnée dans le document pour qu’elle soit utilisable en test (fin de la compréhension du sujet)

1h implémentation du code d’un seul arrêt

1h : implémenter les méthodes pour ajouter ou supprimer des enfants.

1h : algorithme de parcours du graphe (Djisktra)

1h : méthode pour shortest durée (tous les arcs ont un poids de 1)

1h : méthode pour fastest durée (on prend en compte le poids des arcs et on calcule juste la durée minimale.)

2h : On donne une heure de départ et foremost durée qui calcule le trajt pour arriver le plus tot possible (on ajoute un arc de départ à fastest durée qui comptera les temps d’attente)