Tashaly Sobnack
Chloé Chamy
Mélusine Puliéro

PSI : Rapport du rendu 2 PSI

- **préciser le login/password du compte administrateur BDD si nécessaire à l'exécution de votre application**

  password: kakawete

  login: root

- **votre lien GitHub et votre pseudo**

  Notre lien GitHub : https://github.com/meowlusine/PSI_projet

  Nos pseudos:

  -meowlusine

  -chloechamy

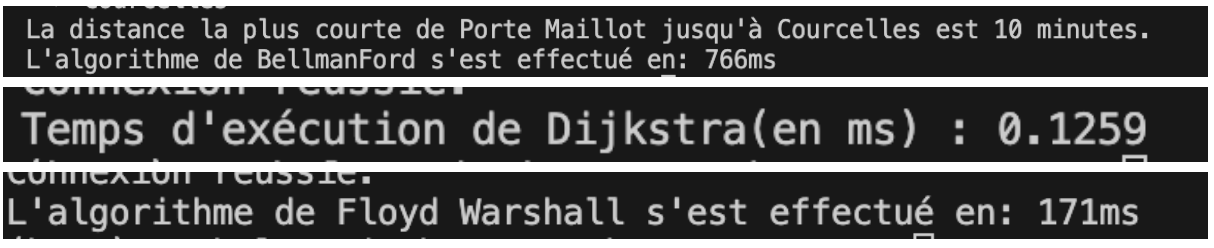
  -Tash-29

- **vos analyses des algorithmes de plus court chemin**

  précision : les algorithmes de Bellman Ford et Floyd Warshall se trouvent dans la classe graphe, et celui de Dijkstra se trouve dans classe algo_chemin

  Pour analyser les algorithmes du plus court chemin, nous utilisons un timer pour connaître le temps d'exécution des algorithmes. Voici nos résultats:

  Nous faisons tourner tous les algorithmes de la station Porte Maillot jusqu'à Courcelles, pour obtenir des résultats comparables



Nous remarquons que l'algorithme le plus rapide est Dijkstra, avec un temps d'exécution de 0.1259 ms, contre 171 ms pour Floyd Warshall et 766 ms pour Bellman Ford. Cependant, le temps d'exécution de Bellman Ford peut être légèrement faussé car nous faisons appel à une fonction TrouverNoeudParId

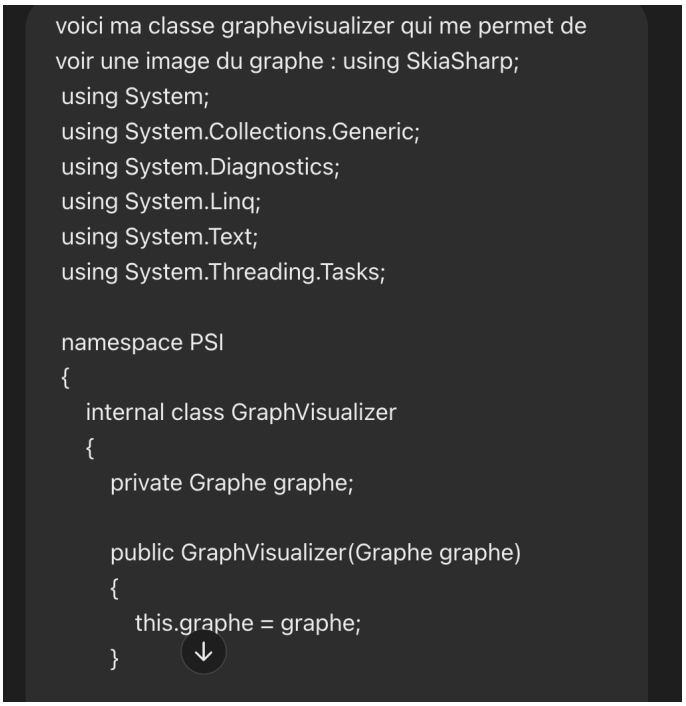
De plus nous savons que la complexité de Bellman Ford est de o(V*E)

et celle de Dijkstra est O (V + E log V)

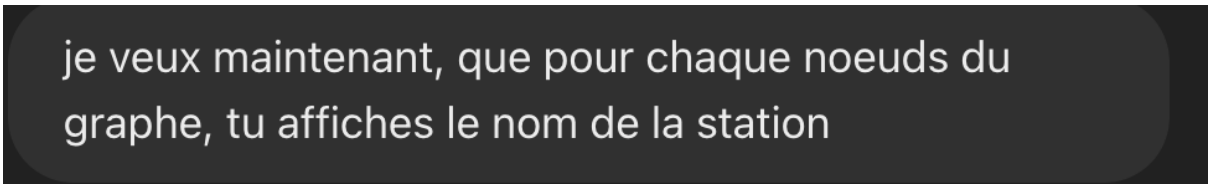
Ce qui est montré par les résultats d'exécution

- **Prompt pour afficher le graphe**

1. Nous lui avons d'abord donné notre ancienne class graphevisualizer



2. **ensuite nous voulions afficher les noms de station**



3. ensuite, nous voulions que les liens soient de la même couleur que la ligne de metro

je veux maintenant que tu changes la couleur des liens. Pour les aretes qui font partie d'une même ligne. Exemple les aretes de la ligne 1 seront toutes jaunes

```
{ "Ligne 1", SKColors.Yellow },
{ "Ligne 2", SKColors.Blue },
{ "Ligne 3", SKColors.Red },
{ "Ligne 4", SKColors.Green },
{ "Ligne 5", SKColors.Orange }
```

4.Enfin nous lui avions demandé de placer les noeuds sur l'image en fonction de leur latitude et leur longitude

maintenant, je veux que tu places sur l'image, les noeuds en fonction de leur longitude et leur latitude

```
var longitudes = graphe.noeuds.Select(n => n.Station.Longitude);
var latitudes = graphe.noeuds.Select(n => n.Station.Latitude);
double minLongitude = longitudes.Min();
double maxLongitude = longitudes.Max();
double minLatitude = latitudes.Min();
double maxLatitude = latitudes.Max();
```