

Documentation CLI pour le fichier main.py

Bienvenue dans la documentation officielle de l'interface en ligne de commande (CLI) pour le fichier main.py. Ce document a pour objectif de vous guider à travers les différentes commandes disponibles, leurs utilisations, et les arguments obligatoires à fournir pour chaque commande.

Introduction

Le fichier main.py est un script Python qui permet d'exécuter diverses opérations sur des graphes, telles que la génération de graphes, le parcours en profondeur (DFS), le parcours en largeur (BFS), la recherche du top influenceur, et bien d'autres. Chaque commande nécessite des arguments spécifiques pour fonctionner correctement.

Commandes disponibles

1. Génération d'un graphe (generate)

Cette commande permet de générer un graphe avec des paramètres spécifiques.

Commande :

```
python main.py generate --oriented|--not_oriented --min_degree <int> --max_degree <int>
--num_communities <int> --max_distance <int> --num_vertices <int> --output <filename>
```

Arguments obligatoires :

- --oriented ou --not_oriented : Spécifie si le graphe est orienté ou non.
- --min_degree <int> : Degré minimum des nœuds.
- --max_degree <int> : Degré maximum des nœuds.
- --num_communities <int> : Nombre de communautés dans le graphe.
- --max_distance <int> : Distance maximale entre les nœuds.
- --num_vertices <int> : Nombre de nœuds dans le graphe.
- --output <filename> : Nom du fichier de sortie où le graphe sera enregistré.

Exemple :

```
python main.py generate --not_oriented --min_degree 2 --max_degree 5 --num_communities 3
--max_distance 10 --num_vertices 100 --output graph.txt
```

2. Chargement d'un graphe (load)

Cette commande permet de charger un graphe à partir d'un fichier.

Commande :

```
python main.py load --filename <filename>
```

Arguments obligatoires :

- `--filename <filename>` : Nom du fichier contenant le graphe à charger.

Exemple :

```
python main.py load --filename graph.txt
```

3. Parcours en profondeur (DFS)

Cette commande permet d'exécuter un parcours en profondeur (DFS) sur le graphe chargé.

Commande :

```
python main.py dfs --root <int>
```

Arguments obligatoires :

- `--root <int>` : Nœud de départ pour le parcours DFS.

Exemple :

```
python main.py dfs --root 0
```

4. Parcours en largeur (BFS)

Cette commande permet d'exécuter un parcours en largeur (BFS) sur le graphe chargé.

Commande :

```
python main.py bfs --root <int>
```

Arguments obligatoires :

- `--root <int>` : Nœud de départ pour le parcours BFS.

Exemple :

```
python main.py bfs --root 0
```

5. Recherche du top influenceur (top-influenceur)

Cette commande permet de trouver le nœud ayant la plus grande influence dans le graphe.

Commande :

```
python main.py top-influenceur
```

Arguments obligatoires :

- Aucun argument supplémentaire n'est nécessaire.

Exemple :

```
python main.py top-influenceur
```

6. Chemin de propagation (propage)

Cette commande permet de trouver le meilleur chemin de propagation entre deux nœuds.

Commande :

```
python main.py propage --start <int> --end <int>
```

Arguments obligatoires :

- --start <int> : Nœud de départ pour la propagation.
- --end <int> : Nœud d'arrivée pour la propagation.

Exemple :

```
python main.py propage --start 0 --end 5
```

7. Temps de propagation (time)

Cette commande permet de calculer le temps de propagation entre deux nœuds.

Commande :

```
python main.py time --start <int> --end <int>
```

Arguments obligatoires :

- --start <int> : Nœud de départ pour le calcul du temps de propagation.
- --end <int> : Nœud d'arrivée pour le calcul du temps de propagation.

Exemple :

```
python main.py time --start 0 --end 5
```

Conclusion

Cette documentation couvre l'ensemble des commandes disponibles dans le fichier main.py. Chaque commande nécessite des arguments spécifiques pour fonctionner correctement. Assurez-vous de fournir tous les arguments obligatoires pour éviter des erreurs lors de l'exécution.

Pour toute question supplémentaire ou en cas de problème, n'hésitez pas à consulter la documentation officielle ou à contacter notre support technique.

Note : Toutes les commandes et arguments sont obligatoires. L'omission d'un seul argument entraînera une erreur et l'interruption du programme.