

Guide d'exécution des commandes en mode CLI

Bienvenue dans ce guide qui vous explique comment exécuter des commandes en mode CLI (Command Line Interface) pour générer des graphes à l'aide du script `generate_graph_test.py`. Nous allons vous guider étape par étape pour que vous puissiez facilement utiliser ces commandes et générer vos propres graphes.

Prérequis

Avant de commencer, assurons-nous que nous avons tout ce qu'il faut :

1. Python 3 : Le script nécessite Python 3 pour fonctionner. Si vous ne l'avez pas encore installé, vous pouvez le télécharger [ici](#).
 2. Le script `generate_graph_test.py` : Assurez-vous d'avoir ce script dans votre répertoire de travail. Si ce n'est pas le cas, récupérez-le auprès de votre équipe ou du dépôt concerné.
 3. Un terminal : On utilisera un terminal pour exécuter les commandes. Sous Windows, vous pouvez utiliser PowerShell ou CMD. Sous macOS ou Linux, utilisez le terminal intégré.
-

Structure de la commande

La commande suit un modèle précis. Voici la structure générale :



```
python3 generate_graph_test.py [--oriented] --num_vertices <nombre> --min_degree <nombre> --max_degree <nombre> --num_communities <nombre> --max_distance <nombre> --output "<nom_fichier>"
```

- `--oriented` (optionnel) : Si on souhaite que le graphe soit orienté, on ajoute cet argument. Sinon, on l'omet.
- `--num_vertices` : Le nombre de sommets dans le graphe.
- `--min_degree` : Le degré minimal d'un sommet.
- `--max_degree` : Le degré maximal d'un sommet.

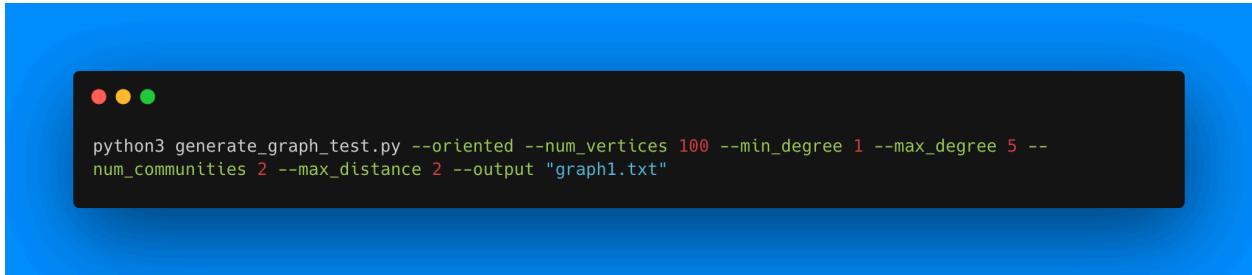
- `--num_communities` : Le nombre de communautés dans le graphe.
 - `--max_distance` : La distance maximale entre deux sommets.
 - `--output` : Le nom du fichier de sortie où le graphe sera enregistré.
-

Exemple d'exécution

Prenons un exemple concret pour mieux comprendre. Supposons que nous voulons générer un graphe avec les caractéristiques suivantes :

- Orientation : Oui (on ajoute `--oriented`).
- Nombre de sommets : 100.
- Degré minimal : 1.
- Degré maximal : 5.
- Nombre de communautés : 2.
- Distance maximale : 2.
- Fichier de sortie : `graph1.txt`.

La commande correspondante sera :



```
python3 generate_graph_test.py --oriented --num_vertices 100 --min_degree 1 --max_degree 5 --  
num_communities 2 --max_distance 2 --output "graph1.txt"
```

Étapes pour exécuter une commande

1. Ouvrir un terminal : On commence par ouvrir un terminal dans notre répertoire de travail.
 2. Taper la commande : On copie ou on tape la commande correspondant au graphe que nous souhaitons générer.
 3. Appuyer sur Entrée : On exécute la commande en appuyant sur la touche Entrée.
 4. Vérifier le résultat : Le graphe est généré et enregistré dans le fichier spécifié (par exemple, `graph1.txt`). On peut ouvrir ce fichier pour vérifier son contenu.
-

Astuces et bonnes pratiques

- Varier les paramètres : N'hésitons pas à jouer avec les paramètres (`--num_vertices`, `--min_degree`, etc.) pour générer des graphes avec des caractéristiques différentes.

- Nommer les fichiers de sortie : On peut donner des noms explicites aux fichiers de sortie pour s'y retrouver plus facilement (par exemple, graphe_100_sommets.txt).
 - Utiliser l'argument --oriented : Si on a besoin d'un graphe orienté, on n'oublie pas d'ajouter cet argument. Sinon, le graphe sera non orienté par défaut.
-

Exemples de commandes

Voici quelques exemples de commandes pour vous inspirer :

1. Générer un graphe non orienté avec 50 sommets :

```
python3 generate_graph_test.py --num_vertices 50 --min_degree 1 --max_degree 4 --num_communities 3 --max_distance 2 --output "graphe_50.txt"
```

2. Générer un graphe orienté avec 200 sommets :

```
python3 generate_graph_test.py --oriented --num_vertices 200 --min_degree 2 --max_degree 8 --num_communities 5 --max_distance 3 --output "graphe_200_oriente.txt"
```

3. Générer un graphe avec 120 sommets et une distance maximale de 4 :

```
python3 generate_graph_test.py --num_vertices 120 --min_degree 3 --max_degree 6 --num_communities 4 --max_distance 4 --output "graphe_120.txt"
```

Exemple de prompt

1. `python3 generate_graph_test.py --oriented --num_vertices 100 --min_degree 1 --max_degree 5 --num_communities 2 --max_distance 2 --output "graph1.txt"`
2. `python3 generate_graph_test.py --num_vertices 50 --min_degree 3 --max_degree 6 --num_communities 4 --max_distance 4 --output "graph2.txt"`
3. `python3 generate_graph_test.py --oriented --num_vertices 200 --min_degree 2 --max_degree 8 --num_communities 5 --max_distance 3 --output "graph3.txt"`
4. `python3 generate_graph_test.py --num_vertices 75 --min_degree 1 --max_degree 4 --num_communities 3 --max_distance 2 --output "graph4.txt"`
5. `python3 generate_graph_test.py --oriented --num_vertices 150 --min_degree 2 --max_degree 7 --num_communities 4 --max_distance 3 --output "graph5.txt"`
6. `python3 generate_graph_test.py --num_vertices 120 --min_degree 3 --max_degree 6 --num_communities 2 --max_distance 4 --output "graph6.txt"`
7. `python3 generate_graph_test.py --oriented --num_vertices 80 --min_degree 1 --max_degree 5 --num_communities 3 --max_distance 2 --output "graph7.txt"`
8. `python3 generate_graph_test.py --num_vertices 300 --min_degree 2 --max_degree 9 --num_communities 6 --max_distance 5 --output "graph8.txt"`
9. `python3 generate_graph_test.py --oriented --num_vertices 90 --min_degree 2 --max_degree 6 --num_communities 3 --max_distance 3 --output "graph9.txt"`
10. `python3 generate_graph_test.py --num_vertices 60 --min_degree 1 --max_degree 4 --num_communities 2 --max_distance 2 --output "graph10.txt"`
11. `python3 generate_graph_test.py --oriented --num_vertices 180 --min_degree 3 --max_degree 7 --num_communities 4 --max_distance 4 --output "graph11.txt"`
12. `python3 generate_graph_test.py --num_vertices 110 --min_degree 2 --max_degree 5 --num_communities 3 --max_distance 3 --output "graph12.txt"`
13. `python3 generate_graph_test.py --oriented --num_vertices 250 --min_degree 2 --max_degree 8 --num_communities 5 --max_distance 4 --output "graph13.txt"`

14. `python3 generate_graph_test.py --num_vertices 70 --min_degree 1 --max_degree 3 --num_communities 2 --max_distance 2 --output "graph14.txt"`
 15. `python3 generate_graph_test.py --oriented --num_vertices 130 --min_degree 2 --max_degree 6 --num_communities 3 --max_distance 3 --output "graph15.txt"`
 16. `python3 generate_graph_test.py --num_vertices 95 --min_degree 3 --max_degree 5 --num_communities 4 --max_distance 2 --output "graph16.txt"`
 17. `python3 generate_graph_test.py --oriented --num_vertices 220 --min_degree 2 --max_degree 7 --num_communities 5 --max_distance 4 --output "graph17.txt"`
 18. `python3 generate_graph_test.py --num_vertices 85 --min_degree 1 --max_degree 4 --num_communities 3 --max_distance 2 --output "graph18.txt"`
 19. `python3 generate_graph_test.py --oriented --num_vertices 160 --min_degree 2 --max_degree 6 --num_communities 4 --max_distance 3 --output "graph19.txt"`
 20. `python3 generate_graph_test.py --num_vertices 140 --min_degree 3 --max_degree 7 --num_communities 3 --max_distance 4 --output "graph20.txt"`
-

Conclusion

Nous avons maintenant toutes les clés en main pour exécuter des commandes en mode CLI et générer des graphes selon nos besoins. N'hésitons pas à expérimenter avec les différents paramètres pour explorer les possibilités offertes par ce script. Si vous avez des questions ou des difficultés, n'hésitez pas à demander de l'aide à votre équipe ou à consulter la documentation du script.

Bonnes expérimentations ! 🚀